1991

Algorithms for blending surface generation.

Filkins, Peter Charles

Springfield, Virginia: Available from National Technical Information Service

http://hdl.handle.net/10945/27938

Algorithms for Blending Surface Generation

by

Peter Charles Filkins

Submitted to the Department of Ocean Engineering
on May 10, 1991, in partial fulfillment of the
requirements for the degrees of
Naval Engineer
and
Master of Science in Mechanical Engineering

Algorithms for Blending Surface Generation

by

Peter Charles Filkins

B.S. in Ocean Engineering (June 1977)
United States Naval Academy

Submitted to the Department of Ocean Engineering
in partial fulfillment of the requirements for the degrees of

Naval Engineer

and

Master of Science in Mechanical Engineering

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 1991

Algorithms for Blending Surface Generation

by

Peter Charles Filkins

Submitted to the Department of Ocean Engineering
on May 10, 1991, in partial fulfillment of the
requirements for the degrees of
Naval Engineer
and
Master of Science in Mechanical Engineering

Abstract

This thesis presents a general algorithm for the representation and approximation of blending surfaces between two non-uniform rational B-spline (NURBS) surfaces using surfaces of the same class. The algorithm permits the approximation of blending surfaces with up to curvature continuous boundary conditions along the intersection of the blending surface with the primary surfaces to be joined. These intersections are called the linkage curves.

The blending surfaces are approximated by low degree B-spline surfaces, created using a lofting method. Isoparameter curves of the approximating blending surface are constructed based on differential geometry properties of the underlying surfaces. First derivative properties are required to satisfy tangent plane continuity boundary conditions. To satisfy curvature continuity conditions, the cross-link curves are constructed to match the curvature of a normal section of the underlying surface at the end point of the cross-link curves on the linkage curves. An adaptive sampling procedure is used to improve the approximation.

Finally, the algorithm is applied to the practical problem of designing a fillet for the intersection of a marine propeller blade with a hub. An adaptation of the method which creates a standard marine propeller fillet automatically is also presented.

Thesis Supervisor: Nicholas M. Patrikalakis
Title: Associate Professor of Ocean Engineering

Acknowledgements

This work would not have been possible without Professor N. M. Patrikalakis, my thesis supervisor, who provided me with much guidance and insight in this field of work, as well as helping me identify most of the important literature. Mr. S. T. Tuohy, who got me started in this work, and who provided countless hours of assistance throughout the development. I would also like to thank Mr. M. S. Drooker, manager of the M.I.T. Ocean Engineering Design Laboratory, and the rest of the Laboratory graduate students.

I would like to thank the U.S. Government and the U.S. Navy, who provided both tuition and salary during my three years at M.I.T.. The experiences I have had here will have a profound impact on my professional life, and I am grateful for having been given this opportunity.

The Design Laboratory work in this area was supported, in part, by the M.I.T Sea Grant College Program and the Naval Sea Systems Command of the U.S. Navy under contract No. NA86AA-D-SG089, and NA16RG0093-01.

I wish to thank my kids, Ryan and Kathleen, who still recognize me with enthusiasm when I come home at night, despite the long days, late nights, and lost weekends. And, finally, I would like to thank my wife, Karen, for her love and support and understanding. Many of the illustrations were prepared, on short notice, by her. Without her support, this work could never have happened.

Contents

1 **Introduction** 9

2 **Survey of Previous Work** 17

2.1 Introduction 17

2.2 Spherical and Circular Blends 18

2.3 Parametric Surface Blending 21

2.3.1 Tangent Plane Continuity Blending Methods 21

2.3.2 Curvature Continuous Blending Methods 25

2.3.3 Partial Differential Equation Blending Methods 27

2.3.4 Energy-Based Variational Blending Methods 29

2.4 Marine Propeller Fillets 31

3 **Formulation of the Blending Problem** 34

3.1 Introduction 34

3.2 Underlying Surfaces 35

3.2.1 Polynomial B-Spline Curves and Surfaces 37

3.2.2 Rational B-Spline Curves and Surfaces 41

3.3 Linkage Curves 42

3.4 Directional Curves 43

3.5 Boundary Conditions 44

3.5.1 Tangent Plane Continuity 45

3.5.2 Curvature Continuity 45

4 Blending Surface Generation **51**

4.1 Introduction 51

4.2 Linkage Curve Approximation 52

4.3 Construction of Cross-Link Curves 55

4.3.1 Position Continuity 57

4.3.2 Tangent Plane Continuity 57

4.3.3 Curvature Continuity 62

4.4 Approximation of the Blending Surface 67

5 Numerical Examples **70**

5.1 Position Continuous Blends 71

5.2 Bias Function Effects 73

5.3 Marine Propeller Fillets 78

5.3.1 Integral B-Spline Surface Fillet 81

5.3.2 3T-T/3 Fillet 81

5.3.3 Pseudo 3T-T/3 Fillet 88

6 Conclusions and Recommendations **95**

A Marine Propeller Geometry **100**

B Creation of a 3T-T/3 Marine Propeller Fillet **103**

B.1 Linkage Curve on the Blade 105

B.2 Fillet Plane 108

B.3 Cross-Link Curves 110

B.3.1 Centers of Circular Arcs 110

B.3.2 Generation of Cross-Link Curves 114

B.4 Approximation of 3T-T/3 Fillet Surface 117

List of Figures

1.1 Components of a blending surface 13

4.1 Tangent plane construction 58
4.2 Possible directions of t 60
4.3 Blend with wrong tangent direction 61
4.4 Tangent vector in terms of surface parameters 65

5.1 Poorly parameterized linkage curves 72
5.2 Satisfactory parameterization check 73
5.3 Constant scalar bias function, $q_j = 0.4$ 74
5.4 Distance function bias factor, $b_j = 2.5$ 75
5.5 Curvature continuous blend, bias factor $b_j = 6$ 77
5.6 Curvature continuous blend, bias factor $b_j = 12$ 78
5.7 Blend with gouging, bias factors $b_1 = 10, b_2 = 3$ 79
5.8 Blend with loop, bias factors $b_j = 3$ 79
5.9 B-spline representation of a propeller blade 82
5.10 Tangent continuous blade-to-hub fillet blend for blade in Figure 5.9 . 83
5.11 Tangent continuous fillet between blade in Figure 5.9 and hub 83
5.12 Curvature continuous blade-to-hub fillet blend for blade in Figure 5.9 85
5.13 Curvature continuous fillet between blade in Figure 5.9 and hub . . . 85
5.14 3T-T/3 fillet blend for blade shown in Figure 5.9 86
5.15 3T-T/3 fillet blend for blade shown in Figure 5.9 87

5.16 3T-T/3 fillet with blade and hub 87
5.17 Propeller blade 89
5.18 3T-T/3 fillet for blade in Figure 5.17 89
5.19 3T-T/3 fillet surface for blade shown in Figure 5.17 90
5.20 Tangent continuous pseudo 3T-T/3 fillet for blade in Figure 5.9 . . . 90
5.21 Tangent continuous pseudo 3T-T/3 fillet for blade in Figure 5.9 . . . 91
5.22 Curvature continuous 3T-T/3 fillet for blade shown in Figure 5.9 . . 93
5.23 Curvature continuous 3T-T/3 fillet for blade shown in Figure 5.9 . . 93

A.1 Propeller blade geometry (from [Kerwin 86]) 101
A.2 Foil section 101

B.1 3T-T/3 fillet cross-link curve geometry 104
B.2 Fillet plane orientation 109
B.3 3T-T/3 detailed geometry 111
B.4 Center of T/3 circle 114

List of Tables

5.1 Curvature continuous fillet with isoparametric linkage curve of Figure 5.13 84
5.2 Tangent continuous pseudo 3T-T/3 fillet 91
5.3 Curvature continuous pseudo 3T-T/3 fillet 92

Chapter 1

Introduction

The shape of objects used in engineering applications reflects both functional requirements and manufacturing processes. Assemblies may be made of parts with simple geometric shapes because of the manner in which they interact, such as shafts and bearing surfaces, or because it is convenient to manufacture them in that manner. On the other hand, complicated sculptured surfaces may be required to satisfy functional requirements such as hydrodynamic or structural performance specifications. There are also aesthetic reasons for choosing a particular shape.

When parts are assembled into the final structure, sharp edge intersections often arise. When these edges and intersections are internal to the assembly, or concave, they can cause stress concentrations or initiate flow disturbances. When they are external projecting features, they can pose safety hazards. To address these problems, a smooth transition surface is usually provided. These secondary surfaces are called blends.

In traditional design and manufacture, blending surfaces occur as the by-product of the limitations of the tools and processes used in their production. As such they are often referred to as engineering surfaces rather than design surfaces. Examples include the minimum radius of a cutting tool in the machining of parts, and forming limitations of casting, forging and pressing processes. If a smooth surface is required,

fabrication guidelines are usually given to achieve the desired effect. Consider, for example, the junction of two machined surfaces. The designer can specify the size of the ball-end mill used to form the junction and can define check surfaces which bound the movement of the cutter. A blending surface of circular cross-section will be produced between the check surface and the machined surface. The disadvantage of this technique is that the designer is unable to completely analyze the performance of the surface because a portion of the geometry of the surface is unavailable in explicit form. Further, the resulting blending surface is an interpretation of the manufacturing instructions, completely dependent on the skill and experience of the machinist. Modern computer aided design and manufacturing technology increasingly emphasizes automation of processes and, therefore, relies on the explicit representation of the entire surface of an object, including secondary blending surfaces.

This thesis examines algorithms for the representation and approximation of blending surfaces. A general blending surface is a surface which provides a smooth transition between different, but intersecting surfaces, or is one which smoothly joins two or more disconnected surfaces. Blending surfaces are often required on edges and at corners and joints. The principal characteristic of a blending surface is that it provides a smooth transition between the surfaces that it joins, and that it joins these surfaces with some specified degree of continuity. The exact shape of the interior of the blending surface is usually less important than the manner in which it joins the principal surfaces.

While some applications simply require the presence of a blending surface, others may have functional requirements, such as the joint between an aircraft wing and the fuselage, the bulbous bow and hull of a ship, or the blade and hub of a marine propeller. When the blending surface plays a part in the ultimate performance of the component, it becomes a design surface which must be accurately specified in the design process. If the surface is left as a traditional by-product type of surface, the designer can not guarantee the performance of his design.

Having established the need for the explicit representation of blending surfaces, the general characteristics of such a method are considered. The method should:

- be applicable to a wide range of geometric shapes and configurations. It must be able to join simple planar surfaces, algebraic surfaces such as quadrics, tori, and surfaces of revolution, and complex free-form surfaces.
- be predictable in its use so that it can be automated. Given that, it must also offer sufficient flexibility so that the designer can manipulate the blending surface in regions where problems arise.
- allow tangent plane continuity along a common boundary between surfaces.
- allow surface curvature continuity, if required, across a common boundary.
- allow comparison of the surface position and differential properties along the common boundary for verifying continuity of the required degree.
- generate blends of a local nature. The method should not alter the global geometry of the surfaces to be joined.
- allow high accuracy approximation to make possible the integration of blending surfaces in modern geometric modellers and to address the need for high accuracy manufacturing operations.

In this thesis, a general method for the representation and approximation of blending surfaces which join two non-uniform rational B-spline (NURBS) surfaces with a surface of the same class is implemented. NURBS surfaces were chosen to represent the primary and the blending surfaces because the form provides a general, unambiguous mathematical representation able to represent a wide range of surfaces. Additionally, there are complete standard data formats for NURBS surfaces which provide for the direct exchange of geometrical information between different geometric modelling, design and manufacturing systems [Smith 88].

To better understand the method, a brief description of some useful terms is provided:

underlying surfaces. Two or more surfaces which are to be joined in a smooth manner. These surfaces are also known as **primary surfaces.**

linkage curve. A curve which defines the extent of the blending surface on the surface to be joined. The linkage curve is the intersection of a primary surface and the blending surface.

cross-link curve. A planar curve which connects corresponding parameter values on each linkage curve. The cross-link curves form a family of curves from which the blending surface is lofted.

directional curve. A curve in three-dimensional space which determines the orientation of the cross-link curves, given the endpoints determined by the linkage curves.

bias functions. A control parameter which the designer may use to influence the shape of the interior of the blend. Bias functions typically affect the first partial derivatives of the blending surface across the linkage curves.

boundary conditions. The degree of continuity with which the cross-link curves and the resulting blending surface join the underlying surfaces.

Figure 1.1 illustrates the relationships of the supporting curves.

The algorithm approximates the blending surface with an integral B-spline surface which is up to curvature continuous to the underlying surfaces. The linkage curves are approximated to within a specified tolerance and the knot vectors are merged. The cross-link curves are created using the common knot vector of the linkage curves, the directional curve, and the parametric partial derivatives of the underlying surfaces, up to second order, if required by the boundary conditions. The cross-link curves are

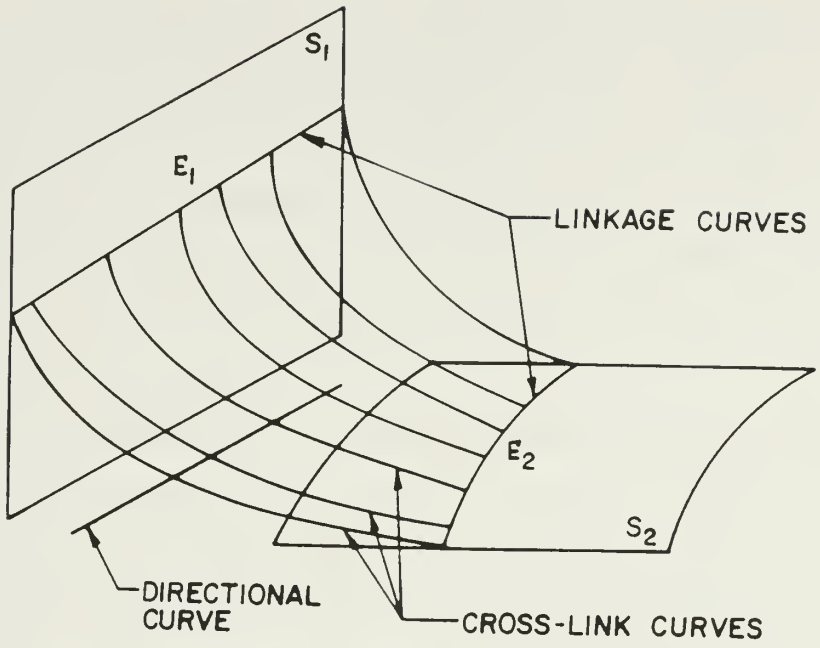


Figure 1.1: Components of a blending surface

then lofted to form the blending surface. An adaptive sampling procedure is used to test the blending surface against specified tolerances. Cross-link curves are added, where necessary, and the surface relofted until the boundary conditions are satisfied.

A significant contribution of the algorithm is the degree of control provided to the designer over the boundary conditions along the linkage curves. Most blending algorithms permit position and tangent plane continuity across the boundaries. Position continuity is important for ensuring the blending surface in fact joins the underlying surfaces. By itself, however, position continuity cannot ensure smooth transition surfaces. Tangent plane continuity is the most common type of boundary condition requirement, and is adequate for most applications. Indeed, [Mortenson 85] observes that most mechanical parts have rounded edges which blend into planar faces for which tangent continuity is sufficient. For more complicated surfaces such as sculptured or free-form surfaces, there can exist significant curvature discontinuities along such a boundary. For aerodynamic and hydrodynamic surfaces, curvature discontinuities can lead to undesirable discontinuites of fluid particle accelerations. The

algorithm given in this thesis allows the accurate approximation of curvature continuous blending surfaces for both simple and complex underlying surfaces. One of the features of the algorithm we developed, contributing to its flexibility, is that the boundary condition along one linkage curve is independent of the boundary condition specified along the other linkage curve. It is possible, for example, to represent a blending surface which is curvature continuous along one boundary while being only position continuous along the other.

The creation of complete geometric models of complex structures encountered in marine applications has been an area of particular interest in the MIT Ocean Engineering Design Laboratory. For an application of the algorithm we examined the creation of blending surfaces between marine propeller blades and hubs. Increasingly demanding performance requirements coupled with more complex shapes are forcing marine propeller manufacturers to use numerically controlled (NC) machining in place of traditional hand-finishing methods. While the use of NC machining has the advantage of repeatability between components, improved performance, and reduced unit production costs, it requires an explicit mathematical representation of the entire component. An area of particular concern in the design and representation of marine propellers is the junction of the propeller blade with the hub. This sharp edge intersection presents both structural and hydrodynamic concerns which are addressed with a feature called a fillet. The fillet is simply a blending surface which contributes to the structural strength of the blade and helps to reduce vibrational blade loading, while providing a smooth transition surface which joins the principal hydrodynamic surfaces to reduce adverse hydrodynamic effects. Several examples of marine propeller fillet blends are given to demonstrate our algorithm.

The thesis is organized as follows.

Chapter 2 presents a survey of previous work in the area of blending surface representation and approximation.

Chapter 3 presents a formulation of the blending surface problem addressed in this

work. The presentation begins with an overview of the basic steps in our algorithm for the design and representation of a blending surface. Various types of surface representation are considered, and our reasons for the use of non-uniform rational B-spline (NURBS) surfaces are given in some detail. A brief review of integral and rational B-spline curve and surface theory is included. The chapter continues with details concerning the specification of linkage curves, directional curves and boundary conditions. As noted above, the algorithm allows boundary conditions along the linkage curves of position, tangent plane and curvature continuity. A discussion of certain differential geometry properties of curves and surfaces necessary for the representation of curvature continuous blending surfaces concludes the chapter.

Chapter 4 describes in detail the implementation of our method, including the approximation of the linkage curves in terms of low-degree B-spline curves, the construction of the cross-link curves, and the approximation of the blending surface by lofting of the cross-link curves. Improvement of the blending surface through adaptive sampling for tangent plane and curvature continuous boundary conditions is discussed.

The method is illustrated with a series of examples in Chapter 5. The examples demonstrate the importance of proper parameterization of the linkage curves. Control of the shape of the blending surface with the bias functions is illustrated, along with two biasing methods. We end with several examples of marine propeller blends, which serve to illustrate the practical applicability of the method in marine propeller design and manufacturing.

The thesis concludes with a summary of the method and recommendations for further work in Chapter 6.

Two appendices are also included.

Appendix A provides general information on marine propeller geometry and definitions of the relevant terminology to aid in understanding some of the examples.

Appendix B gives the details for our implementation of an automatic blending

technique for the standard 3T-T/3 marine propeller blade-to-hub fillet.

Chapter 2

Survey of Previous Work

2.1 Introduction

Various techniques for generating blending surfaces have been reported in the literature. [Woodwark87] presents a thorough survey of earlier methods, the simplest of which involve superficial blends. These methods consist mainly of adding attributes to the description of an intersection between two surfaces. Since there is no explicit representation of the blending surface, the methods have limited design value and, in general, are only useful in providing simple manufacturing instructions. Early techniques which do provide explicit representation of the blend surface are polyhedral blending methods. Smooth edges are created on polyhedrally-defined objects by using recursive subdivision methods to chop-off corners, or by using weighting factors to make local topological changes. These are repatching techniques where parts of a surface are replaced with smaller patches. Surface blending techniques extend these methods and involve the generation of free-form surfaces to smoothly join other surfaces. The original surfaces may or may not intersect. The techniques are primarily implicit formulations based either on Liming's method of cross-sectional design using conic sections [Faux 79], or on potential methods such as that given by [Hoffmann 87c]. Volumetric blending techniques are common to Constructive

Solid Geometry (CSG) systems and Boundary-Representation (B-Rep) solid modellers [Rossignac 84] [Rossignac 85]. These methods involve the creation of new surface geometry and restructure the model to incorporate topological properties to permit further set operations on the object. The models may be either parametric or implicit.

The following sections will discuss in some detail several current methods proposed for blending surface generation. The discussion begins with spherical and circular blending techniques, including the rolling ball blend, a popular technique for describing blending surfaces generated during machining (Section 2.2). Parametric surface blending techniques which more easily deal with sculptured and free-form surfaces appear in Section 2.3. The techniques are further divided into tangent plane continuity methods (Section 2.3.1) and methods offering higher order continuity (Section 2.3.2). Recent work in representing blending surfaces as solutions to boundary value problems posed in terms of partial differential equations is reviewed (Section 2.3.3), as well as energy-based variational methods of free-form surface design applicable to the blending problem (Section 2.3.4). Finally, as a particular example of parametric surface blending, some information is presented on the design of blade-to-hub fillets for rational B-spline (NURBS) representations of marine propellers (Section 2.4).

2.2 Spherical and Circular Blends

As noted in Chapter 1, blending surfaces have traditionally been the by-product of manufacture, not design. Several techniques appear in the literature to model blends which would be produced by ball end mills or disk cutters on numerically controlled (NC) machines. If one can model the blend in this way one might then hope to design and represent it in a more explicit fashion.

[Pegna 87] and [Pegna 88] present methods for spherical and circular blends based on sweep theory. Kinematically, the blend might be represented as the envelope of

a sphere moving while in contact with the underlying surfaces. The analogy is then extended to sweeping a circular arc. The method is an application of generalized cylinders, and is derived from rigid sweep theory, a class of three-dimensional variable sweep geometric models [Pegna 87]. A swept surface is created by a planar curve or cross section, fixed in the normal plane of the Frenet frame of a curve [Farin 88], which is then carried along a twice differentiable space curve without stationary points or points of zero curvature. The cross section curve is called the generatrix and the curve which defines the frame's behavior in space is the directrix. Reduced rigid sweeps are derived from this basic swept surface model by relaxing the requirement that the generatrix be planar. The frame's motion is contrained to be a type of rigid body motion known as Darboux's motion, a class of motion which occurs naturally in the study of systems of orthogonal curvilinear coordinates. Darboux's motions constrains the rotation of the Frenet frame as the origin of the frame translates along the directrix. A more general class, defined as rigid sweeps, decouples the rotation of the frame from its position, thus relaxing the Darboux rigid body motion requirement. This technique defines the motion of the frame with two directrices, one to describe the translation of the origin of the frame, and the other to give the orientation of the frame relative to the translational trajectory. To complete the description, rigid sweeps may have a variable generatrix, one which deforms as the frame moves.

[Pegna 88] proposes several blending techniques using rigid sweeps. These techniques are referred to as either spherical or circular blends with constant or variable radius. Constant radius spherical blending describes the envelope of a sphere with constant radius R moving in contact with the surfaces. The path traced by the center of the sphere is the directrix and is the intersection of the normal offset surfaces of the two underlying surfaces at the same offset distance. The generatrix is an arc of a great circle of the sphere. The same surface can be generated using a constant radius circular blend. For a circular blend, representing a disk cutter, the orientation of the disk must be controlled to maintain tangency with the underlying surfaces.

This additional control represents the difference in a blend machined with a ball end mill on a three axis machine and that of a disk cutter on a four or five axis machine. Additional blending methods proposed allow the radius of the sphere or disk to vary. The methods require the directrix to be C^1 continuous and that it be equidistant from both underlying surfaces. For variable radius spherical blending this distance is the radius of the sphere. [Pegna 87] provides detailed results of these blending techniques and closed form solutions where possible. The blends produced ensure continuity of position and tangent plane along the linkage curves. The implementation is procedural since it requires the evaluation of the intersection of the offset surfaces to find the directrix, and the orthogonal projection of points onto the underlying surfaces to find the boundaries of the generatrix.

[Sanglikar 90] present a mathematical model for the rolling ball blend. The blended surface as well as the underlying surfaces are parametric surfaces. The approach is called modulation and is similar to the spherical and circular sweep blends of [Pegna 87] and [Pegna 88]. The technique splits the blending problem into the design of two spatial curves, a spine and a cross section or pattern curve which correspond to the directrix and generatrix, respectively. To form the modulated surface, the pattern curve is dragged or swept along the spine. The pattern curve may or may not deform as it moves. If the pattern curve is planar, the orientation relative to the tangent of the spine may be varied to satisfy different boundary conditions or other user-specified relationships. As with Pegna's methods, the surfaces to be blended are offset a distance R along the normal vectors to the surfaces, with each surface offset the same distance. The intersection of these surfaces forms the spine of the blend surface, and traces the path of the center of the rolling ball of radius R. Since the offset surfaces are defined in terms of the underlying surface parameters, the intersection curve can also be expressed in terms of these parameters.

To define the pattern curve, a planar curve is generated which represents an arc which is a portion of a great circle of the rolling sphere tangent at its endpoints to

the underlying surface. The set of resulting points of tangency represent the linkage curves referred to in other blending methods. Linkage curves, however, are never explicitly defined. Rather, the method generates a family of planar pattern curves, each of which is circular, of radius R, and of variable arc length. A mathematical expression is then derived for the blend surface, expressing the patch in terms of the spine curve parameter, the normal vectors to the surface at the linkage point expressed in terms of the spine parameter, and a shape factor h, $0 < h < R$. For sculptured surfaces, however, the resultant expression may be of high degree, although not as high as an equivalent implicit model. As with [Pegna 87] and [Pegna 88], the method is procedural because it requires the evaluation of an offset surface intersection.

2.3 Parametric Surface Blending

2.3.1 Tangent Plane Continuity Blending Methods

[Varady 89] discusses a technique for blending parametric surfaces which has been implemented in the solid modeling system FFSolid. The system operates on Boundary-Representation (B-Rep) models and uses quadratic curves and double quadratic surfaces. The blending technique is implemented as one of a set of operations that may be performed on a model and generates an explicit blend surface. The blending surface may be operated on by any of the other solid model operations, including reblending operations.

The technique involves the replacement of an edge or a sequence of edges by a face or sequence of faces. This creates a smooth surface which joins the faces on either side of the replaced edge. The set of operations includes the creation of linkage curves on the faces adjacent to the edge, and then trimming the surfaces to the linkage curve. The blending surface is generated by fitting a surface joining the underlying surfaces at the linkage curves with position and tangent continuity. When a sequence of edges is being blended, the technique permits sharp edges such as those which arise at a

corner to remain or the edge may be smoothed. To create linkage curves a series of points on each surface are specified. An adaptive quadratic curve fitting technique is then used. The series of points may be selected to reproduce many types of blends, including pseudo-rolling ball blends. In general, a point on the edge is selected and the plane normal to the edge at that point is constructed. Points on the surfaces can then be identified which also lie in this plane and are at a specified distance from the point on the edge. The distance can be constant or a function of the position of the point along the edge, and can be measured either as arc length along the intersection curve of the plane and surface, or as an offset distance from the point on the edge. A sorted data base is created which provides a position and corresponding unit tangent vector for every point relative to a base datum. The linkage curve is then fit to the data points using a least square quadratic curve segment fitting algorithm.

To complete the blend, quadratic curve segments are generated between corresponding points on the linkage curves using either a normal plane or an arc-length correspondence. This set of segments makes up a double quadratic strip, a lengthwise double quadratic surface one patch wide, and explicitly defines the blend. The blending surface is defined using the same type of geometric elements used by the other operations in the B-Rep modeller. All solid model operations may be performed on the blend surface, including reblending.

[Bardis 89] presents a technique for constructing tangent continuous blends between non-uniform rational B-spline (NURBS) surface patches, using either open or periodic linkage curves. The method approximates the blending surface by a non-uniform, bicubic integral B-spline patch which interpolates position and normal vector on the linkage curves at an equal number of points on both patches. They use two methods to construct linkage curves. The first method constructs rational B-splines in each patch parameter space by offsetting the intersection of the patches on each patch using a variable distance along the geodesics (see section 3.3).

Another method involves specification of the linkage curves arbitrarily by the

user as proposed by [Hansmann 87]. A further variant implemented is a combination of the these two techniques. Once defined, the linkage curves are approximated in three-dimensional space with a common knot vector. Using this same knot vector, the unit normal vectors to each linkage curve, which are at the same time tangent to the respective surface, are approximated by integral cubic B-spline curves. The approximation technique is an iterative process which ensures convergence to within user-specified tolerances. The resulting blend surface matches position and tangent plane continuity along the linkage curves. The shape of the blend surface may be controlled with user-specified bias functions. The bias functions are applied to the unit normal vectors to the linkage curves, which also lie on the tangent plane of the primary surfaces, and thus control the position of the control polygon vertices.

[Koparkar 91] presents a technique of curve modulation for the blending of parametric surfaces. Curve modulation is similar to rigid sweep theory. The important features of general modulated surfaces are discussed with the citation for rolling ball blends by [Sanglikar 90]. [Koparkar 91] modifies the modulation technique to design an appropriate pattern curve at each selected point as one traces along the directrix. The pattern curve is designed to satisfy position and tangent plane continuity requirements on the underlying surfaces. The tangent directions are selected as either perpendicular to the linkage curve, or tangent to the surface in a plane established by corresponding points on the two linkage and the directrix. The parameterization selected for the pattern curve is a rational form of a conic arc. A conic shape factor is used to control the blending surface as either elliptic, parabolic, or hyperbolic. The shape factor can further be made a function of the directrix parameter w such that the blend varies along the directrix.

Two techniques are proposed for the design of the directrix and the linkage curves. In the first method, the surfaces to be blended are offset along the surface normal vectors a distance R_1 and R_2. The intersections of the offset surfaces form the directrix. This is similar to the spherical and circular blending methods [Sanglikar 90].

[Pegna 87], and [Pegna 88] but the offset distances R_1 and R_2 are allowed to differ. In the special case where R_1 and R_2 are equal and the pattern curve is circular, a spherical or circular blend in achieved. The linkage curve is derived automatically using a geometric correspondence between the normal offset surfaces and the underlying surfaces. A point (u, v) on an underlying surface is mapped to the same (u, v) parameters on the normal offset surface. The intersection of the two offset surfaces is a curve which can therefore be expressed in terms of the parameter values of either surface. The linkage curve is defined as the mapping of the intersection curve onto the underlying surface using the correspondence of the common parameters.

The automatic generation of the directrix and linkage curves from the user-defined offsets at distances R_1 and R_2 may not be sufficient for more complicated applications. To provide the designer with control over the shape and position of the linkage curves, a method employing fanout surfaces is used. Unlike the offset surface, which depends only on the original surface, the fanout surface depends on both the surface and the linkage curve on that surface. The fanout surface is constructed as a ruled surface described by

$$\mathbf{P}^*(s,t) = \mathbf{P}(u(s), v(s)) + t \cdot \hat{\mathbf{P}}(u(s), v(s))$$

where $\mathbf{P}^*$ is a point on the fanout surface with the parameters s and t, $\mathbf{P}$ is a point on the underlying surface with parameters $u(s), v(s)$ describing the linkage curve in the parameter space of the surface, and $\hat{\mathbf{P}}$ is the unit normal vector to the surface at $\mathbf{P}$. By holding the parameter value s constant and varying t, isoparameter lines along the surface normal vectors are obtained. The other set of isoparameter lines is formed by holding t constant and varying s. Two fanout surfaces are constructed from the user-specified linkage curves from both surfaces. Once again, the spine is the intersection of these surfaces. The linkage curves are the orthogonal projections of the directrix onto the underlying surfaces.

The methods presented by [Koparkar 91] are procedural. The offsetting technique works well for simple blends where automatic linkage curve generation is acceptable.

For more complex geometry the fanout surface method permits greater designer control but at the expense of greater computational complexity.

[Tuohy 91b] and [Tuohy 91c] present a method for the approximation of a blending surface as an integral bicubic B-spline surface which is tangent plane continuous with the underlying surfaces along the linkage curves. The method requires linkage curves specified on both surfaces and a third space curve called a directional curve used to control the orientation of the construction plane of the cross-link curves. The cross-link curve establishes an isoparametric correspondence between the linkage curves, and is similar to the generatrix or pattern curve in other cited methods. The linkage curves are first approximated with integral cubic B-splines curves. The cross-link curves are then constructed such that they are position and tangent plane continuous to the underlying surfaces. The cross-link curve also lies in a plane containing a point on each linkage curve and a point on the directional curve. The magnitude of the tangent vectors at the ends of the cross-link curve is controlled with bias functions. The bias functions control the shape of the cross-link curve and can be defined independently for each boundary. Once the cross-link curves are defined, they are lofted to create a rational B-spline surface. The process is iterative and converges to a user-specified tolerance.

2.3.2 Curvature Continuous Blending Methods

[Hansmann 85] and [Hansmann 87] present the fundamentals of a general, interactive method for the design of smooth transitions or free-form blending surfaces between general surfaces with curvature continuity. As examples of free-form surfaces, integral cubic tensor product B-spline patches are used. For analytically defined surfaces, quadric surfaces parameterized in terms of transcendental functions are used. Linkage curves are cubic B-splines interactively defined in the parameter space of the surfaces to be joined. Differential geometry is used to develop tangent plane and normal curvature properties of the primary surfaces along the linkage curves and in a direction

orthogonal to these curves. These properties, along with the linkage curves, provide the boundary conditions for the connecting surface which is a lofted parametric surface using cubic B-spline blending functions. The technique permits up to curvature continuous transitions across the blending surface boundary. The blending surface must be treated as a procedural surface, however, since the surface is irrational with radicals of high degree polynomials.

[Pegna 89a] presents a criterion for guaranteeing second order smoothness or curvature continuity for blending surfaces. Curvature continuity across a linkage curve generally means that the normal curvatures agree in every direction at every point along the linkage curve. There are many ways of formulating this situation, and some of these are discussed in Chapter 3. The contribution in the paper by Pegna and Wolter is the Linkage Curve Theorem. This theorem states that two surfaces joined with first order smoothness, or normal continuity, along a first order continuous linkage curve can be shown to be second order smooth if the normal curvatures on each surface agree in one direction other than the tangent direction to the linkage curve. There is no requirement on the direction other than it not be tangent to the linkage curve. Furthermore, since curvature continuity is achieved by matching normal curvatures in just one direction, the problem of constructing second order smooth blending surfaces is effectively reduced in dimension to one of curve fairing.

Using these results, [Pegna 89] presents a method for designing blending surfaces which ensure curvature continuity. The method is an extension of Pegna's work with rigid sweeps cited earlier [Pegna 87] [Pegna 88]. Here, a space curve is introduced as the directrix and is the axis of a rigid sweep. The linkage curves are constructed as orthogonal projections of the directrix onto the underlying surfaces. An isoparametric mapping is used to relate the two linkage curves and the directrix. Because the linkage curves are derived from the directrix the problem of undesirable twist in the blend caused by poor parameterization of independently designed linkage curves is reduced. To create the blend a swept surface is used with a generatrix which has endpoints on

the linkage curves, end tangents parallel to the surface tangents in the plane formed by the isoparameter mapping, and end curvature such that the normal curvature of the underlying surface and the blend is identical at the endpoint, again in the direction formed by the isoparameter mapping. As examples, a generatrix of Bézier form is used, with third, fourth and fifth order solutions demonstrated. The fourth and fifth order solutions have design parameters to control the steepness of the blend at the linkage curves. [Pegna 89] shows that the blend is curvature continuous if the directrix and the underlying surfaces are second order continuous and the design parameters are first order continuous functions of the directrix parameterization. Furthermore, the isoparametric mapping ensures the end tangent vectors to the generatrix are not tangent to the direction of the linkage curve, thus satisfying the Linkage Curve Theorem of [Pegna 89a] to ensure curvature continuity.

2.3.3 Partial Differential Equation Blending Methods

A method for the representation of surfaces as the solution to partial differential equations and its application to the creation of blending surfaces are described in [Bloor 89a] and [Bloor 89b]. The blend is defined as a function which solves a boundary value problem. The boundary conditions are typically the position of the blending surface on the underlying surfaces (the linkage curves), and a number of derivatives on the boundary as determined by the degree of continuity required between the blend and the underlying surfaces. The blending surface is represented by the parametric function $X = X(u, v)$ where X is the solution to the partial differential equation

$$L^M_{u,v}(X) = F(u, v)$$

where $L^M_{u,v}$ is a partial differential operator of order M in the independent parameter values u, v. The vector valued function F is generally zero for blend surfaces. The class of operators used are elliptic partial differential equations (PDE). The equations,

an example of which is the minimal surface equation, are well-behaved, smoothing functions which generally possess unique solutions without local oscillations. For blending surface design this leads to smooth surfaces without self-intersections or inappropriate intersections with the underlying surfaces.

In [Bloor 89a], a second order PDE is used to create a position continuous blend. To control the shape of the blend between the linkage curves a design parameter is added to the PDE. The parameter permits varying the influence or the smoothing effect of the operators in the two parametric directions. To achieve tangent plane continuity a fourth-order PDE is given, also with a design parameter for control. Higher order continuity is suggested (a sixth-order PDE for curvature continuity) but not demonstrated. The examples presented involve analytic solutions involving transcendental functions and as such have limited use in sculptured surface design. Numerical solutions are also examined and these are expected to have a greater range of applicability.

In [Bloor 89b], the role of the design parameters in the PDEs is examined more closely. The nature of the elliptic PDE allows derivation of closed form solutions to the equations provided the boundaries are properly selected. Design parameters included in the solution permit the designer to control the magnitude and direction of the tangent vector at the boundary. A smoothing parameter controls the relative smoothing of the dependent variables between the parametric directions. The effect is to control the extent of propagation of boundary effects into the interior of the surface. Both design parameters can be either constants or functions of the surface parameters. As functions they can be specified at either or both boundaries, and can be either in phase or out of phase.

In [Bloor 89c], the method is extended to the generation of n-sided patches. [Woodwark87] briefly discusses topological problems associated with sculptured surface engineering. Many common surfaces cannot be represented by a regular array of rectangular patches. A polygonal hole will normally appear somewhere in the surface.

Techniques which subdivide patches to solve this problem tend to relocate the hole, but it still appears. The hole generally has n sides, where n is an integer not equal to 4. As before, the blending surface is the solution to a boundary value problem with elliptic PDEs. The significant adaptation is the parameterization of an n-sided region with two parameters. The boundary conditions are carefully selected to include more than one edge in Euclidean three-dimensional space into the isoparameter edges of the two-dimensional mapping. Junctions are represented by singular points with certain characteristics depending on the behavior of the underlying surfaces and their intersections. The set of equations requires a numerical solution. The blending surface which results joins multiple surfaces and edges with a single parametric patch with no self-intersections, and, depending on the order of the PDE, satisfies any required degree of continuity at the boundaries.

[Bloor 90] shows the PDE method may also be used for the design of free-form surfaces. In this case the boundary conditions are used to manipulate the surface since the designer is free from boundary and continuity conditions imposed on him by the blending surface problem. Internal surface features which cannot be created from the boundary data directly can be introduced by subdivision of the surface, introducing a new boundary to provide the desired feature. An example is the hard chine of a ship hull. A successful implementation of the PDE method would require an interface which translates a less than intuitive PDE design parameter manipulation into a natural design activity. In the case of blending surface design, the method may be useful for the automatic generation of blends.

2.3.4 Energy-Based Variational Blending Methods

[Celniker 89] proposes a method of free-form surface design using energy-based surface models. The method takes advantage of the physical fact that objects deform, when subjected to loads and constraints, to a shape which minimizes the stored energy of the deformation. In this respect, the method is similar to PDE methods. Surface shape

is controlled by boundary conditions, external loads distributed on the surface, and an energy functional. Functionals are integrals which assume maximum or minimum values and are the basis of the calculus of variations [Fox 87]. For energy-based surface models, an energy functional is specified which depends on surface shape. [Celniker 89] implements the technique with parametric surfaces of the form $\mathbf{x}(\mathbf{a}) = [x_1(\mathbf{a}), x_2(\mathbf{a}), x_3(\mathbf{a})]$, where $\mathbf{a} = [u, v]$. The mapping is allowed to change with time, t, such that the shape of the surface is described by $\mathbf{x} = \mathbf{x}(\mathbf{a}, t)$. The energy functional selected in the work depends on the shape of the surface and yields a scalar value for the energy of deformation.

Energy-based methods seek to minimize the internal energy of a system. By defining a functional based on surface curvature and elongation, surfaces deform to naturally fair shapes. Boundary conditions and position constraints may be applied to any point on the surface. The choice and number of constraints applied at a point give the user the means to define precisely the shape of the surface in both a local and global manner. The shape of regions with free points (points with no or few constraints) is controlled by application of forces or loads. The implementation uses forces such as gravity, springs, and pressure as physically meaningful forcing functions. Surface fairness is maintained under external loading by the internal forces of the surface. Internal forces are found by taking the first variation of the energy functional and setting it equal to zero.

The control of shape by minimization of surface energy might be applied to the design of blending surfaces by constructing a parametric surface between linkage curves on the surfaces to be joined, and applying appropriate boundary conditions. The shape of the interior of the blend could be easily controlled by pressure loading.

2.4 Marine Propeller Fillets

Marine propellers represent a particularly complex application of sculptured surface design. The junction of the propeller blade with the hub presents both structural and hydrodynamic concerns. The design of the propeller blade fillet is a particular example illustrated in this research.

[Nittel 89] presents a method for creating a propeller blade fillet. A number of planar sections are specified which are perpendicular to a "nose-to-tail" line, defined as a straight line in three-dimensional space connecting the leading edge to the trailing edge. The planar sections are also parallel to the y-axis of the coordinate system (see Appendix A for standard propeller geometry definitions). This is somewhat unusual since it does not take advantage of the radial nature of the propeller design. The complexity of the blade determines the number of sections required.

Cross-link curves are faired through a small radius fillet and a large radius fillet at points of tangency with the hub surface, the blade surface and with each other. The cross-link curves form a set of splines which are then merged with the splines representing the rest of the propeller blade for use in determining numerically-controlled (NC) machine tool paths and cutter instructions.

[Sabol 83] presents an analytic model for the design of propeller fillets based on the standard 3T-T/3 fillet developed at Philadelphia Naval Shipyard. The model was developed to provide a complete analytic definition of a blending surface which matches as closely as possible the standard 3T-T/3 geometry. A complete analytic definition of the fillet which is consistent with the definition of the blade surface is required for NC machining. To develop the blade-to-hub fillet, a fillet plane is established which includes the normal vector to the surface at the intersection of the propeller blade with the hub, and a point located along one of the lofting lines of the propeller blade (see [Davis 70] for an explanation of the blade lofting procedure). The point on the lofting line is called fillet-to-blade intersection, and it is located a distance, T, from the intersection of the blade and the hub. The T dimension is the

local thickness and is a function of the maximum thickness of the root section and a shape factor based on position along the blade-to-hub intersection curve. The value of T is computed for a particular parameter u along the blade-to-hub intersection and used to locate the endpoints of the cross-link curve on both the blade and hub surfaces. A system of equations is solved to generate a set of points which represent the cross-link curves. The equations have four initial conditions which define the position and tangent directions at the endpoints of the cross-link curve in the fillet plane. The endpoint on the blade is found along the blade lofting line at the distance T described above. The endpoint on the hub and the tangent directions are found using non-dimensionalized ratios based on the fillet to blade intersection. The ratios were found experimentally to closely approximate the 3T-T/3 fillet. An iterative process is used to find the endpoints on the two surfaces and equations are given to solve for the tangent directions. Once these four initial conditions are known, the system of equations is solved to represent the fillet by point data. The method is general in that it makes no requirements on the geometry of the intersecting surfaces. It is therefore not restricted to cylindrical or conical hub surfaces.

The work on feature extraction from B-spline marine propeller representations [Patrikalakis 90a] is now mentioned briefly. To implement a 3T-T/3 fillet for this thesis, a method to compute the maximum thickness of the hub-to-blade root section was required. In [Patrikalakis 90a], a two-dimensional parametric curve is fit to the set of points where cylinders coaxial to the propeller axis of rotation intersect the blade section. The camber line of the curve is then generated using a system of non-linear differential equations. The cylinder selected for use in this thesis corresponds to the hub. It is shown that there is a critical point on the camber line and that the maximum thickness of the blade section occurs at that critical point. Line segments can be constructed from the suction face to the pressure face of the blade, and the tangent vectors along the hub-to-blade intersection determined. The critical point is the point on the camber line where the segment normal to the camber line connects

points whose tangent vectors are parallel to each other. Taking into account the procedure used to construct blade sections and the form of the thickness function which exhibits a single maximum value, the propeller blade section has, in general, only one critical point. By finding the critical point on the camber line it is therefore possible to compute the maximum thickness of the section.

Chapter 3

Formulation of the Blending Problem

3.1 Introduction

The blending surface problem is the generation of a smooth transition surface between two or more underlying surfaces. The blending surface must match user-defined boundary conditions along user-defined boundaries. The user must be able to control the interior shape of the blend. Finally, the surface must be of a form which permits interrogation of the surface in and exchange between geometric modelling systems. The tasks required to generate a blending surface are:

1. specification of underlying surfaces
2. specification of linkage curves
3. specification of a directional curve
4. specification of boundary conditions
5. construction of cross-link curves
6. approximation of blending surface

7. improvement of approximation

The specification of the problem will be described in this chapter. The design of the blending surface will be demonstrated in Chapter 4.

3.2 Underlying Surfaces

As noted in the Introduction (Chapter 1), smooth transitional surfaces are frequently required in design problems, whether it be for functional reasons or merely aesthetic ones. The utility of an algorithm for the creation of these secondary blending surfaces lies, in part, in the types of primary or underlying surfaces the algorithm is able to process. Our goal in this work was to specify the underlying surfaces as non-uniform rational B-spline (NURBS) surfaces.

[Hansmann 87] introduced the fundamentals of a general system for the design of blending surfaces between primary surfaces of different mathematical representation. To demonstrate the method, he used bicubic tensor product B-spline patches as examples of free-form surfaces. To reduce computational effort, the B-spline functions were formulated as third order polynomials instead of using recursive evaluation [de Boor 78](see equations 3.6 and 3.7 in Section 3.2.1). A second class of surfaces considered were analytically defined quadric surfaces defined in parametric form in terms of transcendental functions. For example, an hyperboloid of one sheet with an implicit definition of

$$F(x,y,z) = \frac{x^2}{r_x^2} + \frac{y^2}{r_y^2} - \frac{z^2}{r_z^2} - 1 = 0 \tag{3.1}$$

is represented parametrically in standard orientation (i.e., at the origin with principal axes aligned with the local coordinate system associated with the surface), by

$$\begin{aligned} x = x(u,v) &= r_x \cos(u)\cosh(v) \\ y = y(u,v) &= r_y \sin(u)\cosh(v) \\ z = z(u,v) &= r_z \sinh(v) \end{aligned} \tag{3.2}$$

with $0 \leq u \leq 2\pi$ and $-\pi \leq v \leq \pi$. At this point it is worth mentioning the differences between implicit and parametric representation. Implicit surfaces represented by a function of the form

$$f(x, y, z) = 0 \tag{3.3}$$

are popular for quadric surfaces. Implicit surfaces have a closed form mathematical representation, and as such operations like intersections and surface interrogation can be performed exactly. On the other hand, surface point evaluation usually requires solution of nonlinear equations. Furthermore, it is difficult for the designer to find a relationship between the algebraic coefficients which define the surface in implicit form and the surface's geometric properties. [Woodwark87] notes that high degree implicit equations produce surfaces with unpredictable behavior, self-intersections, numerical instability, and other related problems. Methods to counteract these problems in the representation of free-form curves and surfaces by implicit, piecewise algebraic functions, defined in terms of triple product B-splines, can be found in [Patrikalakis 88d]. For polynomial (algebraic) surfaces represented in implicit form, [Hoffmann 87c] shows implicit polynomial blending surface constructions are possible using potential methods.

Parametric surface representation is of the form

$$\mathbf{S}(u, v) = [x(u, v), y(u, v), z(u, v)] \tag{3.4}$$

where x, y, and z are the three Cartesian coordinates of a point on the surface. Points on parametric curves and surfaces can be easily computed. Parametric representations lend themselves to efficient computations for graphic display and for numerical control purposes in NC machining [Faux 79]. The use of vector quantities in parametric equations aids in matrix transformations such as translations and rotations. It also allows direct control of the surface. In the case of blending surface creation, parametric surface representations are usually approximations only. The approxi-

mation is controlled by tolerances. Parametric polynomials can represent free-form curves and surfaces easily. Since many primary surfaces will be free-form parametric, a parametric surface is also a natural choice for the blend. For generality, parametric surfaces were selected for the underlying surface and blend representation.

Non-uniform rational B-spline (NURBS) surface representation was chosen for the underlying primary surfaces. The form was chosen because it provides a complete, unambiguous mathematical representation able to represent complex sculptured surfaces as well as the classical algebraic surfaces such as quadrics, tori, and surfaces of revolution with rational polynomial profiles. Conic sections and quadric surfaces can be represented exactly with rational B-splines while retaining the properties of integral polynomial B-splines [Tiller 83]. Finally, there exist complete standard data formats for NURBS surface representations [Smith 88] which provide for the direct exchange of geometrical information between different geometric modelling systems, and design and manufacturing systems.

A brief review of polynomial and rational B-spline theory necessary for this work follows. See also [Alourdas 89], [Patrikalakis 89a] and [Tuohy 91b].

3.2.1 Polynomial B-Spline Curves and Surfaces

A spatial, integral B-spline curve has the form

$$\mathbf{R}_M(t) = \sum_{i=0}^{n-1} \mathbf{P}_i N_{i,M}(t) \tag{3.5}$$

with $\mathbf{P}$ the control polygon of n vertices $\mathbf{P}_i$, $0 \leq i \leq n-1$, and a knot vector $T = (t_0, t_1, \ldots, t_k)$, a set of real numbers such that $t_i \leq t_{i+1}$, $0 \leq i \leq k$. The $N_{i,M}(t)$ are scalar valued piecewise polynomials in the parameter t, and are the basis B-spline functions evaluated recursively with the Cox-de Boor algorithm [de Boor 72][Cox 72]:

$$N_{i,M}(t) = \frac{t - t_i}{t_{i+M-1} - t_i} N_{i,M-1}(t) + \frac{t_{i+M} - t}{t_{i+M} - t_{i+1}} N_{i+1,M-1}(t) \tag{3.6}$$

$$N_{i,1}(t) = \begin{cases} 1 & \text{if } t_i \le t \le t_{i+1} \\ 0 & \text{otherwise} \end{cases} \tag{3.7}$$

The convention $0/0 = 0$ is also assumed. Equation (3.5) represents a piecewise, continuous polynomial of degree $M-1$, order M. The curve is continuously differentiable up to order $M-2$. If a curve of order M has no multiple interior knots it is $M-2$ differentiable. If there are multiple interior knots the continuity order at such knots may be reduced.

The basis functions possess the following properties:

$$N_{i,M}(t) \ge 0 \quad , \quad 0 \le i \le n-1 \tag{3.8}$$

$$\sum_{i=0}^{n-1} N_{i,M}(t) = 1 \quad , \quad t_0 \le t \le t_k \tag{3.9}$$

Equations (3.8) and (3.9) form the basis for the convex hull property. Each interval $[t_i, t_{i+1}]$ of the curve is enclosed within the convex hull of the consecutive M vertices affecting the interval. The entire curve is contained within the convex hull of the defining polygon vertices.

The B-spline approximation to a vector-valued, continuous function $\mathrm{g}(t)$ in the interval $[t_0, t_k]$ is given by

$$V\mathrm{g} = \sum_{i=0}^{n-1} \mathrm{g}(\xi_i) N_{i,M}(t) \tag{3.10}$$

The ξ_i are the nodes of the B-spline basis over the knot vector T and are given by

$$\xi_i = \frac{1}{M-1}(t_{i+1} + t_{i+2} + \cdots + t_{i+M-1}), \quad 0 \le i \le n-1 \tag{3.11}$$

The parametric derivatives of a B-spline curve defined by (3.5) are [de Boor 78]

$$\frac{d\mathbf{R}_M(t)}{dt} = (M-1) \sum_{i=1}^{n-1} \frac{\mathbf{P}_i - \mathbf{P}_{i-1}}{t_{i+M-1} - t_i} N_{i,M-1}(t) \tag{3.12}$$

$$\frac{d^2\mathbf{R}_M(t)}{dt^2} = (M-1)(M-2)\sum_{i=2}^{n-1}\left(\frac{\mathbf{P}_i - \mathbf{P}_{i-1}}{t_{i+M-1} - t_i} - \frac{\mathbf{P}_{i-1} - \mathbf{P}_{i-2}}{t_{i+M-2} - t_{i-1}}\right) \cdot \frac{1}{t_{i+M-2} - t_i} N_{i,M-2}(t) \tag{3.13}$$

For the same B-spline curve, there exist an infinite number of defining polygons with more than the minimum number of vertices. The process of finding an alternative representation by the addition of new knots to the knot vector is called knot refinement or subdivision. Consider a B-spline curve of order M with n control polygon vertices, as given by (3.5). The objective is to define the same curve exactly with a new knot vector,

$$S = (s_0, s_1, \ldots, s_{m+M-1})$$

obtained from the original knot vector

$$T = (t_0, t_1, \ldots, t_{n+M-1})$$

by the addition of p new knots, $(\rho_0, \ldots, \rho_{p-1})$ such that $m = n + p$. The new curve is given by

$$\mathbf{S}_M(t) = \sum_{j=0}^{m-1} \mathbf{C}_j B_{j,M}(t) \tag{3.14}$$

where the control polygon vertices $\mathbf{C}_j$ are such that $\mathbf{R}_M(t) = \mathbf{S}_M(t)$. A general method for knot refinement applicable to any B-spline curve is the Oslo algorithm [Cohen 80]. By the algorithm, the new vertices $\mathbf{C}_j$ are

$$\mathbf{C}_j = \sum_{i=0}^{n-1} \alpha_{i,j}^k \mathbf{P}_i \qquad \begin{matrix} 0 \le i \le n-1 \\ o \le j \le m-1 \end{matrix} \tag{3.15}$$

The coefficients $\alpha_{i,j}^k$ are given by the recursion relation

$$\alpha_{i,j}^1 = \begin{cases} 1 & \text{if } t_i \le s_j \le t_{i+1} \\ 0 & \text{otherwise} \end{cases} \tag{3.16}$$

$$\alpha_{i,j}^{k} = \frac{s_{j+k-1} - t_i}{t_{i+k-1} - t_i}\alpha_{i,j}^{k-1} + \frac{t_{i+k} - s_{i+k-1}}{t_{i+k} - t_{i+1}}\alpha_{i+1,j}^{k-1}. \tag{3.17}$$

An application of subdivision is the splitting of a B-spline curve at some arbitrary point. A new knot can be added at a point with multiplicity M, and the curve can be considered to consist of two B-spline curves which, taken together, represent the original curve.

A tensor product B-spline surface is a vector-valued function of two parameters, say u, v. The control polyhedron is now topologically a rectangular net of vertices $\mathbf{P}_{i,j}$, $0 \leq i \leq m-1$, $0 \leq j \leq n-1$. The two knot vectors

$$T = (t_0, t_1, \ldots, t_{m+M-1}) \tag{3.18}$$

$$S = (s_0, s_1, \ldots, s_{n+N-1}) \tag{3.19}$$

are associated with the parameters u, v. The integral B-spline surface formulation of order M in the u direction, N in the v direction, is given by

$$\mathbf{P}_{M,N}(u,v) = \sum_{i=0}^{m-1}\sum_{j=0}^{n-1}\mathbf{P}_{i,j}B_{i,M}(u)B_{j,N}(v) \tag{3.20}$$

By application of equations (3.12) and (3.13) the parametric derivatives of the B-spline surface patch with knot vectors (3.18), (3.19) are

$$\mathbf{P}^{\bullet}_{M,N}(u,v) = (N-1)\sum_{i=0}^{m-1}\sum_{j=1}^{n-1}\frac{\mathbf{P}_{i,j} - \mathbf{P}_{i,j-1}}{s_{j+N-1} - s_j}B_{i,M}(u)B_{i,N-1}(v) \tag{3.21}$$

$$\mathbf{P}^{\bullet\bullet}_{M,N}(u,v) = (N-1)(N-2)\sum_{i=0}^{m-1}\sum_{j=2}^{n-1}\left[\frac{\mathbf{P}_{i,j} - \mathbf{P}_{i,j-1}}{s_{j+N-1} - s_j} - \frac{\mathbf{P}_{i,j-1} - \mathbf{P}_{i,j-2}}{s_{j+N-2} - s_{j-1}}\right] \cdot \frac{1}{s_{j+N-2} - s_j}B_{i,M}(u)B_{j,N-2}(v) \tag{3.22}$$

$$\mathbf{P}'_{M,N}(u,v) = (M-1)\sum_{i=1}^{m-1}\sum_{j=0}^{n-1}\frac{\mathbf{P}_{i,j} - \mathbf{P}_{i-1,j}}{t_{i+M-1} - t_i}B_{i,M-1}(u)B_{j,N}(v) \tag{3.23}$$

$$\mathbf{P}''_{M,N}(u,v) = (M-1)(M-2)\sum_{i=2}^{m-1}\sum_{j=0}^{n-1}\left[\frac{\mathbf{P}_{i,j}-\mathbf{P}_{i-1,j}}{t_{i+M-1}-t_i}-\frac{\mathbf{P}_{i-1,j}-\mathbf{P}_{i-2,j}}{t_{i+M-2}-t_{i-1}}\right]\cdot \frac{1}{t_{i+M-2}-t_i}B_{i,M-2}(u)B_{j,N}(v) \tag{3.24}$$

$$\mathbf{P}'^{\bullet}_{M,N}(u,v) = \mathbf{P}^{\bullet\prime}_{M,N} = (M-1)(N-1)\sum_{i=1}^{m-1}\sum_{j=1}^{n-1}[\mathbf{P}_{i,j}-\mathbf{P}_{i,j-1}-\mathbf{P}_{i-1,j}+\mathbf{P}_{i-1,j-1}]\cdot \frac{B_{i,M-1}(u)B_{j,N-1}(v)}{(t_{i+M-1}-t_i)(y_{s+N-1}-s_j)} \tag{3.25}$$

The primes signify differentiation with respect to u, and the dot with respect to v.

3.2.2 Rational B-Spline Curves and Surfaces

Rational B-spline curves and surfaces make use of homogeneous coordinates. For example, if a point $\mathbf{P} = (x, y, z)$ is defined in 3-dimensional Euclidean space, the corresponding homogeneous coordinate is $\mathbf{P}^* = (wx, wy, wz, w)$, where w is the homogeneous coordinate. We assume $w > 0$. The polynomial formulation of the rational B-spline curve in homogeneous space is

$$\mathbf{R}^h_M(t) = \sum_{i=0}^{n-1}\mathbf{P}^h_i B_{i,M}(t) \tag{3.26}$$

The rational B-spline curve is the projection of the curve defined in (3.26) into 3-dimensional Euclidean space

$$\mathbf{R}_M(t) = \frac{\sum_{i=0}^{n-1} w_i\mathbf{P}_i B_{i,M}(t)}{\sum_{i=0}^{n-1} w_i B_{i,M}(t)} \tag{3.27}$$

Notice that by taking $w_i = 1$ and using (3.9), (3.27) reduces to an integral B-spline curve.

The generalization to the rational B-spline surface follows:

$$\mathbf{S}_{M,N}(u,v) = \frac{\sum_{i=0}^{n-1}\sum_{j=0}^{m-1} w_{i,j}\mathbf{P}_{i,j}B_{i,M}(u)B_{j,N}(v)}{\sum_{i=0}^{n-1}\sum_{j=0}^{m-1} w_{i,j}B_{i,M}(u)B_{j,N}(v)} \tag{3.28}$$

The homogeneous coordinates $w_{i,j}$ are again taken to be positive. As with (3.27), if $w_{i,j} = 1$ the integral B-spline surface patch equation is recovered.

[Tiller 83] notes that the properties of integral (polynomial) B-spline curves and surfaces are easily extended to the rational formulation.

3.3 Linkage Curves

The boundary of the blending surface on the underlying surfaces is the linkage curve or trim line. The term linkage curve was introduced by [Hansmann 87] and will be used in this thesis. There are many ways to specify the linkage curve but most are variations of the following:

- interactively selecting points on the underlying surface (typically by specifying them in parameter space and then mapping them to 3-dimensional space), and then fitting a curve through these points in the parameter space of the surface [Hansmann 87]. This method does not require that the surfaces intersect.

- specifying points on the underlying surface to simulate rolling ball or spherical blends and fitting the points with a space curve [Varady 89][Pegna 88] [Pegna 90].

- specifying an offset curve from the intersection of two surfaces, either along geodesic paths on the surface or along isoparameter lines at a constant distance [Patrikalakis 89b] [Bardis 89], or as a function of the parameter of the intersection curve [Sabol 83]. (The geodesic path is locally the path of minimum distance between two points on a surface among all paths lying on the surface

between these points [do Carmo 76]). While the methods are limited by requiring the surfaces to intersect, the blending of non-intersecting surfaces does not often occur in engineering problems.

Linkage curves which are not isoparameter lines will generally be of high degree. Depending on the method used to define the curves, the linkage curves will be either high degree B-splines, procedural curves, or curves defined in the parameter space of the underlying surfaces. No matter what method is used to specify the linkage curve, care must be taken to ensure the curve does, in fact, lie on the surface with sufficiently high accuracy.

The linkage curves must be properly parameterized. When constructing the cross-link curves it is necessary to establish some type of correspondence between the linkage curves. The usual method for establishing this is by a correspondence of equal parameter values. If the linkage curves are not carefully parameterized, however, there is a possibility that this algebraic isoparametric correspondence will link inappropriate points causing undesirable twist and intersections [Koparkar 91] [Pegna 89]. Interestingly, [Koparkar 91] suggests that the parameterization of the underlying surfaces and the linkage curves is unimportant. A suitable blend requires the proper spatial relationship between the curves and the underlying surfaces. To ensure proper correspondence, he uses Euclidean projections from the intersection curve of the normal or fanout surfaces onto the underlying surfaces to establish a geometric correspondence between the linkage curves.

3.4 Directional Curves

Most of the blending surface design methods construct the blend surface with a family of cross-link curves. The cross-link curves usually connect the linkage curves through a specified correspondence scheme and are typically designed as planar curves. The use of a directional curve is one method of defining the orientation of this planar

curve. The directional curve may be defined as:

- the intersection curve of the surfaces to be joined,
- the intersection of offset surfaces or fanout surfaces of the underlying surfaces.
- a free-form space curve specified by the designer to achieve the desired effect both at the boundary and interior to the blending surface.
- the orthogonal projection of one linkage curve onto the other surface.

The usual means of using the directional curve is through an algebraic correspondence of parameters between the linkage curves and the directional curve. The technique essentially determines the direction of tangent vectors to the surface at the linkage point. Alternative techniques are to use orthogonal projections of the directional curve (in this case the directrix) onto the underlying surface, or to establish a geometric correspondence between a Euclidean projection of the directional curve onto the surface.

An alternative to a directional curve is to define a function which determines the direction of the tangent vectors at the linkage point. [Hansmann 87] requires the cross-link curves to be orthogonal to the linkage curve. In an automatic marine propeller fillet example demonstrated in Chapter 5, the cross-link curves are constructed in "fillet planes." Each fillet plane contains a vector normal to the propeller blade surface at the intersection of the blade and hub, and a point on an isoparameter curve extending along the blade surface from the hub to the tip (see Section B.2). No directional curve is required.

3.5 Boundary Conditions

Boundary conditions specify the degree of continuity between the blend surface and the underlying surface along the linkage curve. The simplest type of boundary condition is position continuity. This simply ensures the edge of the blend surface does

in fact lie on the underlying surface along the linkage curve, to within the specified tolerance. No further conditions are required of the blend.

3.5.1 Tangent Plane Continuity

Most blending algorithms require, beyond positional, tangent plane continuity across the linkage curve. [Pegna 89a] refers to this as the first order blending problem. To satisfy tangent plane continuity, the tangent planes to the blend and the underlying surface must match at all points along the linkage curve. This condition is satisfied during construction of the cross-link curves. It may be automatic such as in the case of circular or spherical sweeps, where the generatrix contacts the surface at a point of tangency. More general methods determine the tangent plane to the underlying surface at a point on the linkage curve, and construct the cross-link curve by initiating the linkage curve in the tangent plane in the direction specified by the directional curve. A bias factor can be applied to control the effect of the tangency condition into the interior of the blend.

3.5.2 Curvature Continuity

Curvature continuity, or second order blending, is achieved when normal curvatures agree in every direction at every point along the linkage curve. Before proceeding with the requirements for curvature continuity, a review of some differential geometry concepts is presented.

3.5.2.1 Review of Differential Geometry and Curvature

We take the curve $\mathbf{u} = \mathbf{u}(t)$ on a parametric surface $\mathbf{r} = \mathbf{r}(u, v)$ where

$$\mathbf{r} = \mathbf{r}(u, v) = \begin{bmatrix} x(u, v) \\ y(u, v) \\ z(u, v) \end{bmatrix} \tag{3.29}$$

represents one of the underlying surfaces in the blending problem, and

$$\mathbf{u} = \mathbf{u}(t) = \begin{bmatrix} u(t) \\ v(t) \end{bmatrix} \tag{3.30}$$

the linkage curve on that surface. A specific point on the linkage curve in three dimensions will be referred to as $\mathbf{r}(t)$.

The unit vector normal to the surface is given by

$$\mathbf{n} = \frac{(\mathbf{r}_u \wedge \mathbf{r}_v)}{|\mathbf{r}_u \wedge \mathbf{r}_v|} \tag{3.31}$$

where $\wedge$ denotes the vector product. A unit tangent vector to the linkage curve, $\mathbf{r}'$, is given by [Faux 79]

$$\mathbf{r}' = \mathbf{r}_u \dot{u} + \mathbf{r}_v \dot{v} = \mathbf{A}\dot{\mathbf{u}} \tag{3.32}$$

where

$$\mathbf{A} = \begin{bmatrix} x_u & x_v \\ y_u & y_v \\ z_u & z_v \end{bmatrix}. \tag{3.33}$$

Recalling that the arc length of a parametric curve is

$$s(t) = \int_{t_0}^{t_1} |\mathbf{r}'| \, dt, \tag{3.34}$$

the squared arc element is

$$ds^2 = |\mathbf{r}'|^2 = (\mathbf{r}_u^2 \dot{u}^2 + 2\mathbf{r}_u \cdot \mathbf{r}_v \dot{u}\,\dot{v} + \mathbf{r}_v^2 \dot{v}^2)\, dt^2 \tag{3.35}$$

or, more simply,

$$ds^2 = (E\dot{u}^2 + 2F\dot{u}\,\dot{v} + G\dot{v}^2)\, dt^2 \tag{3.36}$$

where

$$E = E(u, v) = \mathbf{r}_u \cdot \mathbf{r}_u \tag{3.37}$$

$$F = F(u, v) = \mathbf{r}_u \cdot \mathbf{r}_v \tag{3.38}$$

$$G = G(u, v) = \mathbf{r}_v \cdot \mathbf{r}_v \tag{3.39}$$

are the elements of the first fundamental form. This is also referred to as the first fundamental matrix or the metric tensor. It is a measure of the distance in three dimensional space that exists between neighboring points on a surface.

The curvature of a curve parameterized by arc length is

$$\kappa = \kappa(s) = |\mathbf{r}''|, \tag{3.40}$$

and in terms of the general parameter t,

$$\kappa = \kappa(t) = \frac{|\mathbf{r}' \wedge \mathbf{r}''|}{|\mathbf{r}'|^3}. \tag{3.41}$$

The radius of curvature is

$$\rho = \frac{1}{\kappa}. \tag{3.42}$$

Using (3.32) for the curve $\mathbf{u}(t) = [u(t), v(t)]$ on the surface $\mathbf{r} = \mathbf{r}(u, v)$, we obtain

$$\mathbf{r}'' = \mathbf{r}_{uu}\dot{u}^2 + 2\mathbf{r}_{uv}\dot{u}\,\dot{v} + \mathbf{r}_{vv}\dot{v}^2 + \mathbf{r}_u\ddot{u} + \mathbf{r}_v\ddot{v} \tag{3.43}$$

Taking the component of (3.43) in the direction of the surface normal $\mathbf{n}$.

$$\mathbf{r}'' \cdot \mathbf{n} = \mathbf{n} \cdot \mathbf{r}_{uu}\dot{u}^2 + 2\mathbf{n} \cdot \mathbf{r}_{uv}\dot{u}\,\dot{v} + \mathbf{n} \cdot \mathbf{r}_{vv}\dot{v}^2 + \mathbf{n} \cdot \mathbf{r}_u\ddot{u} + \mathbf{n} \cdot \mathbf{r}_v\ddot{v} \tag{3.44}$$

and recalling that $\mathbf{n} \cdot \mathbf{r}_u = \mathbf{n} \cdot \mathbf{r}_v = 0$, we find the second fundamental form or tensor

$$\mathbf{b} = L\dot{u}^2 + 2M\dot{u}\,\dot{v} + N\dot{v}^2 \tag{3.45}$$

where

$$L = L(u, v) = \mathbf{n} \cdot \mathbf{r}_{uu} \tag{3.46}$$

$$M = M(u, v) = \mathbf{n} \cdot \mathbf{r}_{uv} \tag{3.47}$$

$$N = N(u, v) = \mathbf{n} \cdot \mathbf{r}_{vv} \tag{3.48}$$

The second fundamental tensor provides a measure of the curvature of a surface at a point. It can be shown that the normal curvature κ_n of the surface at a point $\mathbf{r}(t)$ along the linkage curve in a given tangent direction $\dot{u}, \dot{v}$ is simply the ratio of the second fundamental form to the first fundamental form,

$$\kappa_n = \frac{L\dot{u}^2 + 2M\dot{u}\,\dot{v} + N\dot{v}^2}{E\dot{u}^2 + 2F\dot{u}\,\dot{v} + G\dot{v}^2} \tag{3.49}$$

A normal section is the plane containing both a unit vector $\mathbf{t} = [\dot{u}, \dot{v}]$ emanating from a point $\mathbf{r}(t)$ in the tangent plane, and the normal vector to the surface at that point. Equation (3.49) therefore gives the normal curvature for any normal section at a point $\mathbf{r}(t)$. [Faux 79] presents the notion of signed curvature for planar curves and for surfaces. For planar curves, the sign of curvature is chosen as required for the application. For surfaces, the curvature is assumed to be positive if the curve of intersection of the normal section with the surface has its center of curvature in the direction of the positive surface normal vector. The principal curvatures are the maximum and minimum values of (3.49), and are denoted κ_1, κ_2. The product $K = \kappa_1\kappa_2$ is the Gaussian curvature, and the mean curvature is $H = \frac{1}{2}(\kappa_1 + \kappa_2)$. The directions corresponding to the principal curvatures are called the principal curvature directions. A line of principal curvature of the surface is a curve on the surface such that, at

all points $\mathbf{r}(t)$ along the curve, the tangent line of the curve is a principal direction. It can be shown that the lines of principal curvature are orthogonal [Farin 88] [do Carmo 76].

An asymptotic direction of the surface $\mathbf{r}(u, v)$ at a point $\mathbf{r}(t)$ is a direction in the tangent plane for which the normal curvature is zero. A geometric interpretation of asymptotic direction is the equation of a conic section known as Dupin's Indicatrix. Knowing the principal curvatures, the normal curvature κ_n of the section formed by the unit tangent vector $\mathbf{t}$ and the surface normal vector is

$$\kappa_n = \kappa_1 \cos^2 \theta + \kappa_2 \sin^2 \theta \tag{3.50}$$

where θ is the angle between the unit tangent vector $\mathbf{t}$ and the direction of the maximum principal curvature. Equation (3.50) is the classical Euler's Formula, expressed in terms of the orthonormal basis formed by the principal curvature directions. Dupin's Indicatrix is the union of conic sections in the tangent plane such that $\kappa_n = \pm 1$. When κ_1 and κ_2 have the same sign at $\mathbf{r}(t)$, the Gaussian curvature is positive, $\mathbf{r}(t)$ is an elliptic point, and Dupin's Indicatrix is the graph of an ellipse. If κ_1 and κ_2 have opposite signs, the point is hyperbolic, and Dupin's Indicatrix gives two hyperbolas with a common set of asymptotic lines. If one of the principal curvatures is zero, the asymptotic lines are parallel straight lines and the point is parabolic.

For an extensive treatment of these concepts refer to [do Carmo 76] and [Farin 88].

3.5.2.2 Conditions for Curvature Continuity

With these concepts in mind, we now consider the problem of curvature continuity of adjoining surfaces. Pegna and Wolter [Pegna 89a] present the following formulations for curvature continuity:

The osculating paraboloids of the underlying surface and the blending surface must match at all points along the linkage curve. The osculating paraboloid for a surface $\mathbf{S}$ with the xy plane as the tangent plane at a point p on $\mathbf{S}$ is given by

[do Carmo 76]:

$$z = \frac{1}{2}(x^2 f_{xx} + 2xy f_{xy} + y^2 f_{yy}) \tag{3.51}$$

where $z = f(x, y)$ is the equation of the surface.

If the blend juncture is already shown to be tangent plane continuous, the following criteria may be used:

1. The Dupin's indicatrices and curvature signs of both surfaces must agree along the linkage curve.

2. The asymptotic directions of the two surfaces agree at all points along the linkage curve.

3. The second fundamental tensors, formed as basis independent, multilinear maps, agree at all points along the linkage curve.

4. The principal curvature directions and principal radii of curvature for the surfaces agree along the linkage curve.

Finally, as noted in Chapter 2, [Pegna 89a] shows that two surfaces which are tangent along a first order smooth linkage curve are curvature continuous if their normal curvature agrees in a direction other than the tangent to the linkage curve for all points along the linkage curve. This formulation is called the Linkage Curve Theorem. Since it is important to our work, this theorem is stated formally from [Pegna 89a]:

Theorem 3.1 *Given two C^2-continuous surfaces $\mathbf{S}_1$ and $\mathbf{S}_2$, and a C^1-continuous linkage curve $\mathbf{E}(t)$ such that $\mathbf{E}' \neq 0$ for all t, $t \in \mathcal{T}$ with $\mathcal{T} \subset \Re$, where $\Re$ is the set of real numbers. The linkage curve represents the tangent plane continuous intersection of the two surfaces. If at some point $\mathbf{E}(t_0) \in \mathbf{E}(\mathcal{T})$ for some tangent direction $\mathbf{X}$ linearly independent from the tangent direction $\mathbf{E}'(t_0)$ the normal curvatures of $\mathbf{S}_1$ and $\mathbf{S}_2$ agree, then the normal curvatures of $\mathbf{S}_1$ and $\mathbf{S}_2$ agree for all tangent directions at the point $\mathbf{E}(t_0)$.*

Chapter 4

Blending Surface Generation

4.1 Introduction

The method for blending surface generation implemented in this thesis approximates a smooth transitional surface as a B-spline surface with user-specified boundary conditions across the linkage curves. The general method is as follows:

1. Approximate the linkage curves with integral cubic B-spline curves. The knot vectors of the approximated linkage curves are then merged to form a basis for the design of the cross-link curves.

2. Construct the family of cross-link curves based on the type of blending surface desired and the specified boundary conditions, linkage and directional curves, and bias functions. The parameter values of the cross-link curves are the nodes of the merged knot vector of the linkage curves.

3. Create the approximating blending surface by lofting the cross-link curves over the merged knot vector of the approximating linkage curves. [Tuohy 91b] shows that by using the merged knot vector and the nodes of the linkage curves, the boundary curves of the blend will be identical to the curves approximating the linkage curves.

4. Evaluate the approximation by checking continuity errors with adaptive sampling of the blending surface and underlying surfaces. The degree of sampling is dependent on the blend continuity specified across the linkage curves.

5. Improve the approximation by the addition of cross-link curves based on the adaptive sampling results until the error is within the specified tolerances.

4.2 Linkage Curve Approximation

The linkage curve defines the extent of the blend on the underlying surfaces. It can be expressed as a B-spline curve, a curve defined in the parameter space of the underlying surface, or as a procedural curve. No matter how the linkage curve is specified, it must in fact lie on the surface to ensure the blending surface joins the underlying surfaces with proper position continuity. We use a method presented by [Wolter 91] and [Tuohy 91b] to approximate the linkage curve with a cubic integral B-spline curve to within position, tangent angle and relative curvature tolerances when compared with the original linkage curve. We assume the linkage curve is expressed as a vector-valued function, $\mathbf{g}(t)$, and seek an approximating integral B-spline curve as in equation (3.5),

$$\mathbf{R}_M(t) = \sum_{j=0}^{n-1} \mathbf{P}_j B_{j,M}(t) \tag{4.1}$$

which interpolates the function $\mathbf{g}(t)$ at the nodes ξ_i of the knot vector

$$T = (t_0, t_1, \ldots, t_{m+M-1}), \tag{4.2}$$

normalized such that the overall parameter interval spans $[0, 1]$. Selecting the nodes as in (3.11),

$$\xi_i = \frac{1}{M-1}(t_{i+1} + t_{i+2} + \cdots + t_{i+M-1}). \tag{4.3}$$

the vertices $\mathbf{P}_i$ of the control polygon of the approximating B-spline may be computed by solving the system

$$\sum_{j=0}^{n-1} \mathbf{P}_j N_{j,M}(\xi_i) = \mathbf{g}(\xi_i), \quad 0 \leq i \leq n-1, \tag{4.4}$$

which can be written in matrix form as

$$[N]\{\mathbf{P}\} = \{\mathbf{g}\} \tag{4.5}$$

where

$$[N] = [n_{ij}]_{0 \leq i \leq n-1, 0 \leq j \leq n-1} \tag{4.6}$$

$$[n_{ij}] = N_{j,M}(\xi_i) \tag{4.7}$$

$$\{\mathbf{P}\} = [\mathbf{P}_0, \ldots, \mathbf{P}_{n-1}]^T \tag{4.8}$$

$$\{\mathbf{g}\} = [\mathbf{g}_0, \ldots, \mathbf{g}_{n-1}]^T \tag{4.9}$$

[de Boor 78] shows that the matrix $[N]$ is non-singular and therefore invertible since we evaluate the system at the nodes of a monotonically increasing knot vector. The system of (4.5) can thus be solved for the control polygon using standard linear algebra techniques.

The initial approximation is then improved by adaptive sampling. The method is explained here because an adaptation of the method is used to improve the blending surface. The approximation is tested at midspan between the nodes for position error using local error bounds. The approximation method ensures the position error at the nodes is zero. [Wolter 91] and [Tuohy 91b] show that the maximum position error occurs near the midspan between two adjacent nodes. The error function, therefore, is simply

$$|\mathbf{R}_M(t_i) - \mathbf{g}(t_i)| < \epsilon_1, \qquad \eta_i = \frac{\xi_i + \xi_{i+1}}{2} \text{ for } 0 \leq i \leq n-2. \tag{4.10}$$

If the position test is satisfied, the angle o is formed by the tangent vector to the approximating B-spline and the tangent to the linkage curve.

$$\left|\arccos\left(\frac{|\mathbf{R}'_M(u_i)\cdot\mathbf{g}'(u_i)|}{|\mathbf{R}'_M(u_i)||\mathbf{g}'(u_i)|}\right)\right| < \epsilon_2, \quad u_i = \frac{\eta_i + \eta_{i+1}}{2} \text{ for } 1 \le i \le n-3. \tag{4.11}$$

where η_i is defined as in (4.10). In practice, it is usually convenient to test the angle difference at the nodes as $u_i \approx \xi_i$ as the distance between successive knots decreases during adaptive sampling.

In similar fashion, relative curvature at the nodes of the approximating curve

$$\kappa = \frac{|\mathbf{r}' \wedge \mathbf{r}''|}{|\mathbf{r}'|^3}$$

is tested such that

$$\left|\frac{\kappa_{\mathbf{g}}(\xi_i) - \kappa_{\mathbf{R}}(\xi_i)}{\kappa_{\mathbf{g}}(\xi_i)}\right| < \epsilon_3. \tag{4.12}$$

For both first and second order tests, the use of geometric properties such as angle or curvature of the approximation and original curve is preferred [Tuohy 91b] since such properties are independent of the curve parameterization. At points where the error test fails, t_i is added to the knot vector T. With this enriched knot vector, a new approximating B-spline curve is constructed. The procedure is repeated until the approximating curve passes all error tests.

From the above discussion, it is noted that improvement of the approximation is restricted solely to knot vector refinement. The shape of any B-spline curve is derived from its control points and the basis functions. The knot vector has a significant influence on the basis functions [Rogers 76] for a given curve order. Since the approximation scheme interpolates the linkage curve at the nodes exactly, knot vector refinement offers the best means of improving the approximation without loosing position accuracy. Furthermore, prudent selection of the initial knot vector will enhance the efficiency of the approximation. Several methods for selection of initial knot vector are available [Hartley 80] [Hoelzle 83] [Hoschek 87].

4.3 Construction of Cross-Link Curves

Two techniques were examined for the construction of cross-link curves. The first algorithm is a general one which creates a family of cross-link curves derived from the linkage curves, the boundary conditions across the linkage curves, a directional curve, and a bias function. The second algorithm is actually a special case of the first, where the linkage curves and cross-link curves are constructed automatically to satisfy specified geometric conditions. The second algorithm is presented as an example in Chapter 5.

The cross-link curves are integral B-spline curves of the form

$$\mathbf{C}(t) = \sum_{i=0}^{m-1} \mathbf{P}_i N_{i,M}(t), \tag{4.13}$$

where M is the order and m the number of vertices of the control polygon, $\mathbf{P}$. The cross-link curves are constructed at the nodes of the common knot vector created by merging the knot vectors of the approximations to the linkage curves. As noted at the beginning of this Chapter, this procedure ensures the boundary curves of the lofted blending surface are identical to the approximating linkage curves. The boundary conditions are used to determine the order of the cross-link curve, the number of control points, and the knot vector. Nine combinations of position, tangent plane and curvature continuity are possible.

The simplest case is position continuity on both underlying surfaces. The cross-link curves are of second order and have two control points with a knot vector of $(0, 0, 1, 1)$. The control points are simply the points on the linkage curves associated with the particular node of the common knot vector being evaluated. This type of cross-link curve is useful as a preliminary check of the linkage curve correspondence.

The next two cases require position continuity on one surface, and position and tangent plane continuity on the other surface. A third order B-spline curve with three

control points,

$$\mathbf{C}(t) = \sum_{i=0}^{2} \mathbf{P}_i N_{i,3}(t), \tag{4.14}$$

is used with the knot vector $(0,0,0,1,1,1)$. These curves are all of Bézier form, a special case of B-spline, and can be raised to cubic B-splines easily [Cohen 85].

The remaining combinations use integral cubic B-splines ($M = 4$). One case requires position and tangent plane continuity on both surfaces. Two other cases require position continuity on one surface, and position, tangent plane and curvature continuity on the other. The cross-link curve for these cases is of the form

$$\mathbf{C}(t) = \sum_{i=0}^{3} \mathbf{P}_i N_{i,4}(t), \tag{4.15}$$

with knot vector $(0,0,0,0,1,1,1,1)$, and is again of Bézier form. The three cases differ from one another only in the manner in which the control points are computed.

Two cases require up to tangent plane continuity on one surface, and up to curvature continuity on the other surface. For these cases a fifth control point is added,

$$\mathbf{C}(t) = \sum_{i=0}^{4} \mathbf{P}_i N_{i,4}(t), \tag{4.16}$$

with knot vector $(0,0,0,0,\frac{1}{3},1,1,1,1)$ or $(0,0,0,0,\frac{2}{3},1,1,1,1)$. For convenience the cross-link curve $\mathbf{C}(t)$ is parameterized such that $\mathbf{C}(0)$ interpolates surface $\mathbf{S}_1$ with linkage curve $\mathbf{E}_1(t)$, and $\mathbf{C}(1)$ interpolates the second surface $\mathbf{S}_2$ with linkage curve $\mathbf{E}_2(t)$. The knot vector is chosen such that the internal knot is closest to the surface with the curvature continuity boundary condition.

The final case requires up to curvature continuity conditions with both underlying surfaces. The cross-link curves now take the form

$$\mathbf{C}(t) = \sum_{i=0}^{5} \mathbf{P}_i N_{i,4}(t), \tag{4.17}$$

with six control points and knot vector $(0,0,0,0,\frac{1}{3},\frac{2}{3},1,1,1,1)$. Having selected the

proper form of the cross-link curve required to satisfy the boundary conditions. the curves are constructed.

4.3.1 Position Continuity

Position continuity is guaranteed to a specified tolerance by evaluating the linkage curves at the nodal points of the common knot vector. We label these points on the linkage curves $\mathbf{E}_j(\xi_i)$, $j = 1, 2$, corresponding to surfaces $\mathbf{S}_j$. A common knot vector T, derived from the approximation of the linkage curves, is evaluated at the nodes ξ_i, as defined in (3.11). The cross-link curve associated with the node ξ_i has the form

$$\mathbf{C}(t) = \sum_{i=0}^{m-1} \mathbf{P}_i N_{i,M}(t) \tag{4.18}$$

To satisfy position continuity, then, we require that

$$\mathbf{P}_0 = \mathbf{E}_1(\xi_i) \tag{4.19}$$

$$\mathbf{P}_{m-1} = \mathbf{E}_2(\xi_i) \tag{4.20}$$

4.3.2 Tangent Plane Continuity

Tangent plane continuity for each cross-link curve requires two control points associated with each boundary. The first establishes position continuity. The second control point is located in the plane tangent to the primary surface so that the cross-link curve is tangent to the surface at the linkage curve. The position of the second control point in the tangent plane is determined by the directional curve and a scalar bias function. This function is used to control the magnitude of the tangent vector of the cross-link curve and thus influence its shape.

The direction of the vector in the tangent plane of the primary surface at $\mathbf{E}_j(\xi_i)$ is based on a technique developed by [Tuohy 91b]. The method is illustrated in Figure 4.1. The direction $\mathbf{t}_j$ lies along the intersection of the plane tangent to the

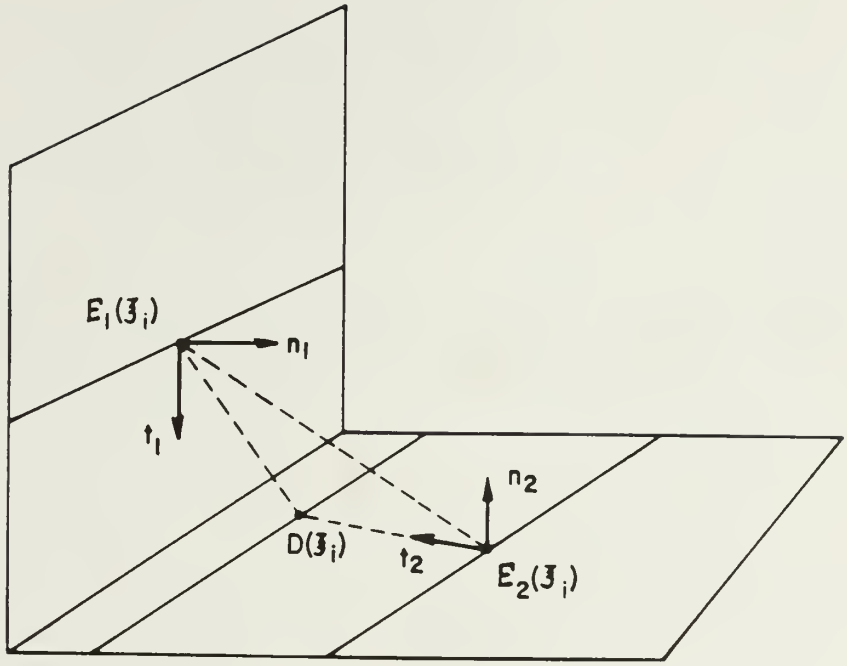


Figure 4.1: Tangent plane construction

primary surface at $\mathbf{E}_j(\xi_i)$, $j = 1, 2$, with a plane containing corresponding points on both linkage curves and the directional curve. The direction is found by computing the cross product of the normal vectors to the two planes.

We first compute the unit normal vector to the plane formed by the points $\overline{\mathbf{E}_1\mathbf{E}_2\mathbf{D}}$, where $\mathbf{E}_1$ and $\mathbf{E}_2$ are corresponding points on the linkage curves and $\mathbf{D}$ is the associated point on the directional curve. The unit normal vector is computed as

$$\mathbf{n}_0 = \frac{(\mathbf{E}_2 - \mathbf{E}_1) \wedge (\mathbf{D} - \mathbf{E}_1)}{|(\mathbf{E}_2 - \mathbf{E}_1) \wedge (\mathbf{D} - \mathbf{E}_1)|} \tag{4.21}$$

The unit normal vector $\mathbf{n}_j$ to the underlying surface $\mathbf{S}_j, j = 1, 2$ is computed as

$$\mathbf{n} = \frac{\mathbf{S}_u \wedge \mathbf{S}_v}{|\mathbf{S}_u \wedge \mathbf{S}_v|}. \tag{4.22}$$

In the event one of the parametric derivatives of the surface is zero, (i.e., a degenerate

patch), the unit normal vector to the surface is found by either

$$\mathbf{n}_j = \frac{\mathbf{S}_{uv} \wedge \mathbf{S}_v}{|\mathbf{S}_{uv} \wedge \mathbf{S}_v|} \quad \text{if } \mathbf{S}_u = 0, \text{ or} \tag{4.23}$$

$$\mathbf{n}_j = \frac{\mathbf{S}_u \wedge \mathbf{S}_{uv}}{|\mathbf{S}_u \wedge \mathbf{S}_{uv}|} \quad \text{if } \mathbf{S}_v = 0 \tag{4.24}$$

as suggested by [Faux 79].

The control points of the cross-link curve required to establish tangent plane continuity at its end point on the linkage curve $\mathbf{E}_1(t)$ are, therefore,

$$\mathbf{P}_0 = \mathbf{E}_1(\xi_i) \tag{4.25}$$

$$\mathbf{P}_1 = \mathbf{P}_0 \pm \mathbf{t}_1 \tag{4.26}$$

$$\text{where} \quad \mathbf{t}_1 = q_1(\mathbf{n}_0 \wedge \mathbf{n}_1) \tag{4.27}$$

where q_1 is a positive bias factor which can depend on ξ_i. The control points for the other end of the cross-link curve associated, with $\mathbf{E}_2$, are

$$\mathbf{P}_{m-1} = \mathbf{E}_2(\xi_i) \tag{4.28}$$

$$\mathbf{P}_{m-2} = \mathbf{P}_{m-1} \pm \mathbf{t}_2 \tag{4.29}$$

$$\text{where} \quad \mathbf{t}_2 = q_2(\mathbf{n}_0 \wedge \mathbf{n}_2) \tag{4.30}$$

where q_2 is also a positive bias factor which can depend on ξ_i, and be defined independently from q_1. The $\pm$ in equations (4.26) and (4.29) provide a way to ensure the tangent vectors are oriented in the proper direction. Depending upon the order in which the cross products are taken and the direction of the surface normal vector, the tangent vector will point either towards the directional curve or away from it while still lying along the intersection of the tangent and surface normal planes (Figure 4.2). Without a directional check, cross-link curves and blending surfaces may arise such as shown in Figure 4.3. To account for this we compute the angle between the tangent

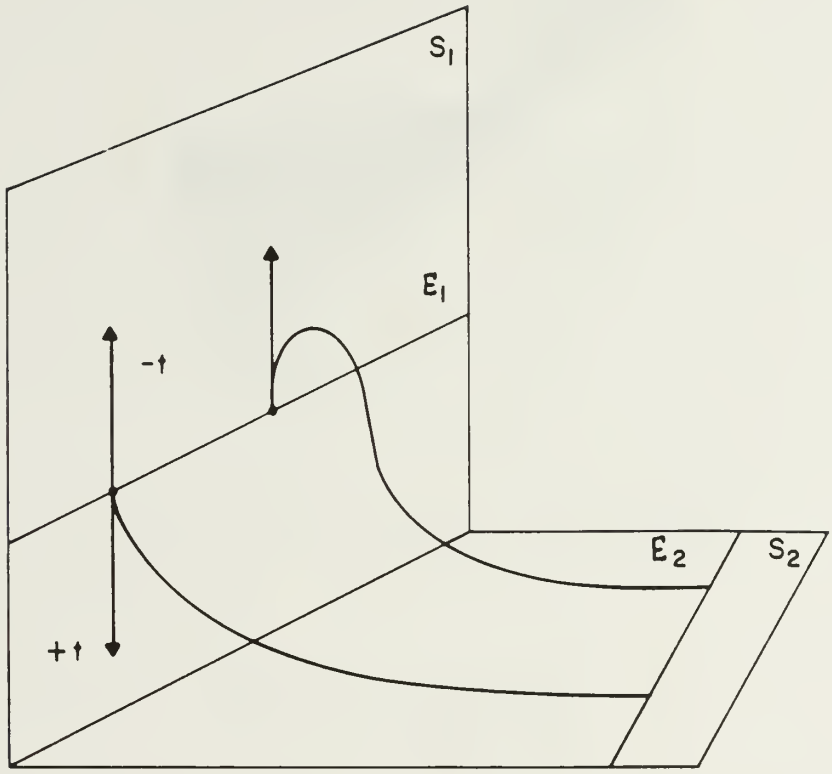


Figure 4.2: Possible directions of t

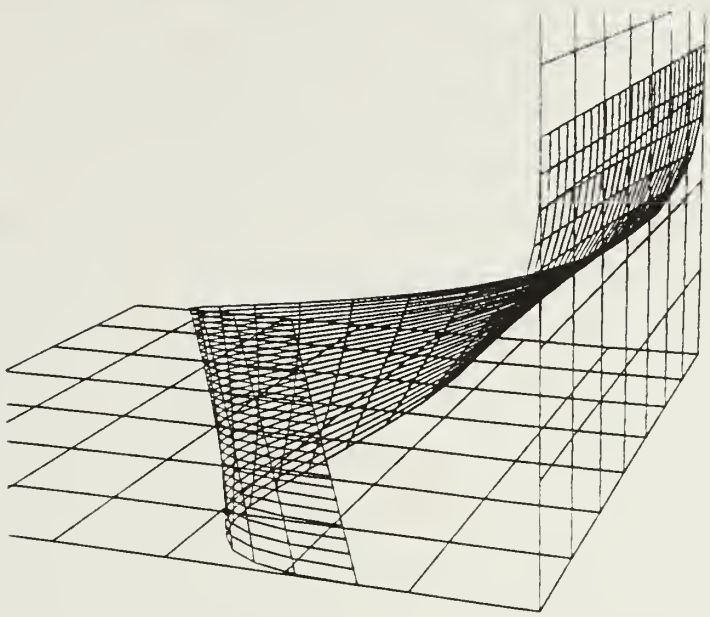


Figure 4.3: Blend with wrong tangent direction

vector $\mathbf{t}_j$ and the vector $\mathbf{D} - \mathbf{E}_j$,

$$\phi_1 = \arccos \frac{\mathbf{t}_j \cdot (\mathbf{D} - \mathbf{E}_j)}{|\mathbf{t}_j||\mathbf{D} - \mathbf{E}_j|}, \tag{4.31}$$

and the angle between the opposite tangent vector, $-\mathbf{t}_j$, and $\mathbf{D} - \mathbf{E}_j$,

$$\phi_2 = \arccos \frac{-\mathbf{t}_j \cdot (\mathbf{D} - \mathbf{E}_j)}{|\mathbf{t}_j||\mathbf{D} - \mathbf{E}_j|}. \tag{4.32}$$

If $\phi_2 < \phi_1$, the tangent vector $-\mathbf{t}_j$ is selected, thus determining the sign in equations (4.26) and (4.29) and ensuring the correct tangent vector is chosen. The benefit of this technique is that the choice of proper tangent direction is automatic and the tangent is guaranteed to point towards the directional curve. Thus, if a blending surface such as that shown in Figure 4.3 is desired, all that is required is that the directional curve be located to select that tangent orientation.

The factor q_j in equations (4.27) and (4.30) represent the scalar bias functions.

The bias function controls the shape of the cross-link curve by controlling the magnitude of the tangent vector. It can be a constant value or a function $q_j(t)$ which varies with the parameterization of the linkage curves or directional curve. Examples of constant and variable bias functions are given in Section 5.2.

The method presented above (equations (4.25) to (4.30)) allow the definition of planar cross-link curves, each of which is in a plane $\overline{\mathbf{E}_1\mathbf{E}_2\mathbf{D}}$ controlled by the parametric correspondence of the linkage and directional curves. This method is at variance with the method developed by [Hansmann 87], who chooses vectors $\mathbf{t}_1$ and $\mathbf{t}_2$ to be orthogonal to the linkage curves (as well as to $\mathbf{n}_1$ and $\mathbf{n}_2$) and does not, therefore, have planar cross-link curves even for up to tangent plane continuous blends. Our experimentation indicates that the construction of planar cross-link curves with tangent vectors at the ends not exclusively controlled by the linkage curves but rather from the directional curve provides easy control of the shape of the blend and good flexibility in the design of complex free-form blends.

4.3.3 Curvature Continuity

To satisfy curvature continuity boundary conditions, we rely on differential geometry properties of the underlying surfaces along the linkage curves (see Section 3.5.2.1).

The cross-link curve is of the form given by equation (4.13). Evaluating this equation and the first and second parametric derivatives given in equations (3.12) and (3.13), we get the following system of equations along $\mathbf{E}_1(t)$:

$$\mathbf{P}_0 = \mathbf{E}_1(\xi_i) \tag{4.33}$$

$$\mathbf{P}_1 = \mathbf{P}_0 + \mathbf{C}'_{\xi_i}(0)\left(\frac{t_M - t_1}{M-1}\right) \tag{4.34}$$

$$\mathbf{P}_2 = \mathbf{P}_0 + \mathbf{C}'_{\xi_i}(0)\left(\frac{t_{m+M-2} + t_{M+1} - t_1 - t_2}{M-1}\right) \cdot \mathbf{C}''_{\xi_i}(0)\left(\frac{(t_M - t_2)(t_{M+1} - t_2)}{(M-1)(M-2)}\right) \tag{4.35}$$

For control points associated with the second surface and $\mathbf{E}_2(t)$.

$$\mathbf{P}_{m-1} = \mathbf{E}_2(\xi_i) \tag{4.36}$$

$$\mathbf{P}_{m-2} = \mathbf{P}_{m-1} - \mathbf{C}'_{\xi_i}(1)\left(\frac{t_{m+M-2} - t_{m-1}}{M-1}\right) \tag{4.37}$$

$$\mathbf{P}_{m-3} = \mathbf{P}_{m-1} - \mathbf{C}'_{\xi_i}(1)\left(\frac{t_{m+M-2} + t_{m+M-3} - t_{m-1} - t_{m-2}}{M-1}\right) \cdot \mathbf{C}''_{\xi_i}(1)\left(\frac{(t_{m+M-3} - t_{m-1})(t_{m+M-3} - t_{m-2})}{(M-1)(M-2)}\right) \tag{4.38}$$

As before, $\mathbf{E}_j(\xi_i)$, $j = 1, 2$ represent the linkage curves evaluated at the nodes ξ_i of the common knot vector, S, of the approximations to the linkage curves, and $\mathbf{C}$ is the corresponding cross-link curve and its parametric derivatives. The subscript ξ_i denotes association with a particular node. The knot vector for the cross-link curve is $T = (t_0, t_1, \ldots, t_{m+M-1})$ as specified in Section 4.2 above. The control points $\mathbf{P}_0, \mathbf{P}_1$ and $\mathbf{P}_{m-1}, \mathbf{P}_{m-2}$ are found using the method shown in Section 4.3.2 to satisfy position and tangent plane continuity. From equations (4.33), (4.34), (4.36) and (4.37), we can therefore solve for $\mathbf{C}'_{\xi_i}(0)$ and $\mathbf{C}'_{\xi_i}(1)$:

$$\mathbf{C}'_{\xi_i}(0) = (\mathbf{P}_1 - \mathbf{P}_0)\frac{M-1}{t_M - t_1} \tag{4.39}$$

$$\mathbf{C}'_{\xi_i}(1) = (\mathbf{P}_{m-1} - \mathbf{P}_{m-2})\frac{M-1}{t_{m+M-2} - t_{m-1}} \tag{4.40}$$

To find the remaining control points $\mathbf{P}_2$ and $\mathbf{P}_{m-3}$, we need information concerning second derivative of the cross-link curve at its end point. We find this using a generalization of the method presented by [Hansmann 87]. Specifically, we obtain the second derivative of the cross-link curve, evaluated at the end points $\mathbf{E}_j(\xi_i)$, from the curvature of the normal section of the underlying surface which contains the cross-link curve tangent vector and the surface normal vector at $\mathbf{E}_j(\xi_i)$.

We first find the curvature of the normal section curve using equation (3.49)

$$\kappa_n(\xi_i) = \frac{L\dot{u}^2 + 2M\dot{u}\dot{v} + N\dot{v}^2}{E\dot{u}^2 + 2F\dot{u}\dot{v} + G\dot{v}^2} \tag{4.41}$$

where E, F, G, L, M, N are the coefficients of the first and second fundamental forms defined in equations (3.37) through (3.39), and (3.46) through (3.48), and $(\dot{u}, \dot{v})$ the direction of the normal section curve in the parameter space of the primary surface. The dot signifies differentiation with respect to the parameter t of the curve on the surface formed by the intersection of the normal section plane and the underlying surface. The underlying surface, of the form

$$\mathbf{S}_{M,N}(u,v) = \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} \mathbf{P}_{i,j} B_{i,M}(u) B_{j,N}(v) \tag{4.42}$$

is evaluated for position, first, second and cross-derivatives recursively using the Cox-de Boor algorithm and equations (3.21) through (3.25). The results are used to solve for the first and second fundamental coefficients.

To find the required tangent direction $(\dot{u}, \dot{v})$ of the normal section in terms of the surface parameters we recall from differential geometry (Figure 4.4 and equation (3.32)) that a nonzero vector $\mathbf{t}$ is tangent to the surface $\mathbf{S}$ at the point $\mathbf{E}(\xi_i)$ if there is a regular curve $\mathbf{r} = \mathbf{r}(t)$ on the surface through $\mathbf{E}(\xi_i)$ such that $\mathbf{t} = \dot{\mathbf{r}}$. If $\mathbf{r} = \mathbf{r}(t) = \mathbf{r}(u(t), v(t))$, then the tangent vector to the surface is

$$\mathbf{t} = \mathbf{r}_u \cdot \dot{u} + \mathbf{r}_v \cdot \dot{v} \tag{4.43}$$

The vectors $\mathbf{r}_u$ and $\mathbf{r}_v$ are the first parametric derivatives of the surface at $\mathbf{E}(\xi_i)$ and are known. We also know $\mathbf{t}$ from the solution to the tangent plane continuity problem in Section 4.3.2. Denoting the components of the vectors as $\mathbf{t} = (t_X, t_Y, t_Z)$, $\mathbf{r}_u = (x_u, y_u, z_u)$ and $\mathbf{r}_v = (x_v, y_v, z_v)$ in the global coordinate system, equation (4.43) can be written in matrix form as

$$\begin{Bmatrix} t_X \\ t_Y \\ t_Z \end{Bmatrix} = \begin{bmatrix} x_u & x_v \\ y_u & y_v \\ z_u & z_v \end{bmatrix} \begin{Bmatrix} \dot{u} \\ \dot{v} \end{Bmatrix} \tag{4.44}$$

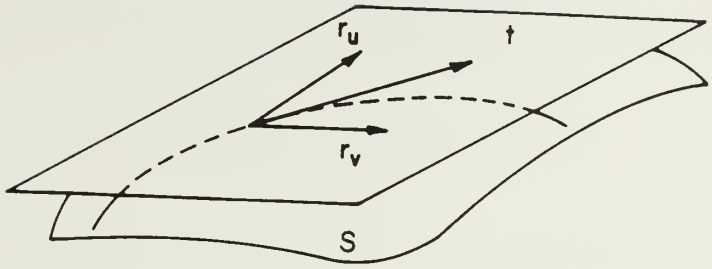


Figure 4.4: Tangent vector in terms of surface parameters

This is a linear system of the form $Ax = b$. We solve for $[\dot{u} \;\; \dot{v}]^T$ using a least squares technique [NAG 89] by taking

$$\mathbf{t} = \mathrm{C}'_{\xi_i}(0) \tag{4.45}$$

and

$$\mathbf{t} = \mathrm{C}'_{\xi_i}(1). \tag{4.46}$$

We then evaluate (4.41) to obtain the normal curvature $\kappa_n(\xi_i)$ of the primary surfaces along the normal section formed by $[\dot{u}, \dot{v}]$.

Having obtained the normal curvature of the primary surfaces at the points $\mathbf{E}_j(\xi_i)$ and directions $\mathbf{t}_j$, $(j = 1, 2)$, we now proceed to construct cross-link curves which have the same curvature at these points and which are also locally normal sections of the blending surface to be generated. Because the tangent plane continuity is already enforced, the relevant normal section planes of the blend at $\mathbf{E}_1$ and $\mathbf{E}_2$ are defined by $(\mathbf{t}_1, \mathbf{n}_1)$ and $(\mathbf{t}_2, \mathbf{n}_2)$, respectively.

To relate the normal curvature to second derivative of the cross-link curves at their end points, we consider a local orthogonal coordinate system with the origin at $\mathbf{E}_j(\xi_i)$, the x-axis parallel to the direction of the tangent vector $\mathbf{t}_j$, the y-axis parallel to the primary surface normal vector $\mathbf{n}_j$, and the z-axis defined by $\mathbf{t}_j \wedge \mathbf{n}_j$. We now compute

the components of the cross-link curve second derivatives at their end points in these two local coordinate systems. We use equation (3.41) for $\mathbf{r} = \mathbf{C}_{\xi_l}(t) = [x_l(t), y_l(t), 0]$ for t near 0 or 1. In this way, we obtain the relation

$$|x_l' y_l'' - y_l' x_l''| = \kappa_n |\mathbf{C}_{\xi_l}'(\tau)|^3 \tag{4.47}$$

for $\tau = 0$ or 1. Using equations (4.45) and (4.46) and decomposing $\mathbf{t}$ along the local axes so that $\mathbf{t} = [t_x, t_y, t_z]$ where $t_z = 0$ and $|\mathbf{t}| = (t_x^2 + t_y^2)^{1/2}$, then equation (4.47) becomes

$$|t_x y_l'' - t_y x_l''| = \kappa_n |\mathbf{t}|^3 \tag{4.48}$$

Equation (4.48) is satisfied by choosing

$$x_l'' = -|\mathbf{t}| \kappa_n \, t_y \tag{4.49}$$

$$y_l'' = |\mathbf{t}| \kappa_n \, t_x \tag{4.50}$$

These components of second derivative $[x_l'', y_l'', 0]$ are then transformed to $\mathbf{C}_{\xi_l}''(0)$ or $\mathbf{C}_{\xi_l}''(1)$, as appropriate, in the global coordinate system using the direction cosines of the axes of the local coordinate system. The final control points $\mathbf{P}_2$ and $\mathbf{P}_{m-3}$ are then solved for using equations (4.35) and (4.38). In this manner, because $\mathbf{C}_{\xi_l}''(0)$ and $\mathbf{C}_{\xi_l}(1)$ are vectors in the planes $(\mathbf{t}_1, \mathbf{n}_1$ and $(\mathbf{t}_2, \mathbf{n}_2)$ the resulting control points will no longer necessarily be contained in the plane $\overline{\mathbf{E}_1\mathbf{E}_2\mathbf{D}}$. Therefore, for curvature continuous blends the resulting cross-link curves are non-planar.

In the implementation of this work described in Chapter 5 and in order to maintain planarity of cross-link curves and their easy control, we developed an alternate method. In this method we impose the curvature values at the ends of the cross-link curves as obtained from normal sections of the primary surfaces in the directions $\mathbf{t}_1$ and $\mathbf{t}_2$. However, we use local coordinate systems containing the plane $\overline{\mathbf{E}_1\mathbf{E}_2\mathbf{D}}$ (i.e., local x-axis in the $\mathbf{t}_j$ direction and local y-axis in the $\mathbf{n}_0 \wedge \mathbf{t}_j$ direction). In this man-

ner, the resulting cross-link curve is planar in its entirety and contained in the plane $\overline{E_1E_2D}$. However, this curve is no longer locally a normal section of the blend at E_1 or E_2 unless $\mathbf{n}_0 \cdot \mathbf{n}_1 = 0$ or $\mathbf{n}_0 \cdot \mathbf{n}_2 = 0$ and, therefore, not all conditions of surface curvature continuity are generally satisfied. This implementation of Chapter 5 resulted in easily controllable blends of smooth form which can be characterized as providing continuity of curvature of cross-link curves.

4.4 Approximation of the Blending Surface

The blending surface is created using a lofting technique where the cross-link curves are lofted using the common knot vector from the merging of the knot vectors of the approximations to the linkage curves. The cross-link curves become isoparameter curves of the blending surface. As noted earlier, use of the common knot vector and cross-link curves at the nodes of the knot vector ensures the boundary curves of the blending surface will be identical with the approximations of the linkage curves.

A technique similar to the improvement of the approximation to the linkage curves is used to improve the accuracy of the blending surface. The surface is sampled at the midspan between the nodes of the common knot vector and checked against tolerances to the specified boundary conditions [Tiller 83]. Position, tangent and curvature continuity along the linkage curve is guaranteed by the lofting technique. To check for tangent plane continuity of the blend, we compute the angle between the unit normal vector to the blending surface, $\mathbf{n}_{b_j}$, along the boundary of the blend with the unit normal vector to the associated underlying surface, $\mathbf{n}_{s_j}$,

$$\phi_j = \arccos\left(\frac{|\mathbf{n}_{b_j} \cdot \mathbf{n}_{s_j}|}{|\mathbf{n}_{b_j}||\mathbf{n}_{s_j}|}\right), \tag{4.51}$$

with $j = 1, 2$, to indicate the underlying surface and linkage curve and b_j the appropriate boundary curve of the blending surface. Sampling is at the midspan between the nodes as defined in (4.10). If the angle found in (4.51) is greater than the specified

tolerance, ϵ_1, the value t_i is added to the common knot vector.

To test for curvature continuity, we extract an isoparameter curve in the cross-link direction from the blending surface, again at the midspan of the nodal points. The isoparameter curve comes from evaluating the surface over the complete range of one parameter maintaining the other parameter constant. The blending surface was constructed such that the cross-link curves become isoparameter curves of the lofted blend surface. Let the blending surface be given by,

$$\mathbf{B}(u,v) = \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} \mathbf{P}_{i,j} B_{i,M}(u) B_{j,N}(v), \tag{4.52}$$

where u, v are the parametric directions along the linkage and cross-link curves. The isoparameter curve extracted from it is of the form

$$\mathbf{I}(v) = \sum_{j=0}^{n-1} \mathbf{R}_j B_{j,N}(v) \tag{4.53}$$

with

$$\mathbf{R}_j = \sum_{i=0}^{M-1} \mathbf{Q}_{i,j} B_{i,M}(t_i). \tag{4.54}$$

We then compute the curvature of the isoparameter curve at the endpoint with a curvature continuity boundary condition.

$$\kappa_I = \frac{|\mathbf{I}'(t) \wedge \mathbf{I}''(t)|}{|\mathbf{I}'(t)|^3}. \tag{4.55}$$

For comparison, we construct a cross-link curve at the same midspan nodal value and compute its curvature, κ_{cl}, in the same manner. The curvatures are then compared to a relative curvature tolerance, ϵ_2, defined as

$$\frac{|\kappa_I - \kappa_{cl}|}{|\kappa_I| + |\kappa_{cl}|}. \tag{4.56}$$

As before, if the test fails the midspan point is added to the knot vector.

The improvement process for curvature continuity compares the blending surface at a point with the cross-link curve which would be added to that point if required. If the new cross-link curve does not improve the surface, it is not added. The improvement of the blend surface tests tangent plane continuity first if tangent plane or curvature continuity is specified. If the tangent plane test fails the point is added to the knot vector and the next interval is sampled. Curvature continuity is only tested if tangent plane continuity is satisfied first. Once all intervals are tested the blending surface is reconstructed with the new knot vector. The process is continued until all continuity tests are satisfied.

As a final remark, we assert that a blending surface lofted from cross-link curves constructed as presented in Section 4.3 while also being normal sections of the blend at $\mathbf{E}_1$ and $\mathbf{E}_2$, is curvature continuous by the Linkage Curve Theorem (Theorem 3.1). The tangent directions at the end points of the cross-link curve are guaranteed to be independent from the tangent direction of the linkage curve for appropriately oriented linkage and directional curves.

Chapter 5

Numerical Examples

We now illustrate the generation of blending surfaces with several numerical examples. The first examples demonstrate just position continuity along the linkage curves. Some examples are given to illustrate some important concepts concerning blending surface shape control with the bias functions. These examples use simple planar and quadratic integral and rational surfaces which permit a clear demonstration of the results. The last sequence of examples illustrate tangent plane and curvature continuous blending surfaces with complex surfaces representing marine propellers.

The routines to compute, interrogate, and display the blending surfaces were developed for use in the MIT Ocean Engineering Design Laboratory geometric modeling system, *Praxiteles*. This system provides an environment for both graphical and computational interaction between a designer and the Silicon Graphics IRIS 4D 120 GTXB workstation. The basic structure of the system is described in [Hottel 89], with more advanced versions documented in [Alourdas 90], [Patrikalakis 90a] and [Hottel 91]. The routines are written in the C programming language with calls to FORTRAN subroutines in the NAG Mark 13 FORTRAN library [NAG 89] where required.

5.1 Position Continuous Blends

Position continuity is the simplest type of blend. The term "blend" is used loosely since the surface generated does not provide a gradual transition between the underlying surfaces but simply connects them. The primary advantage of the position continuous blending surface is that it is created very quickly. In our implementation, a preliminary position continuous blending surface is constructed after specification and approximation of the linkage curves to verify proper parameterization of the linkage curves. If the linkage curves are designed independently there is a possibility of introducing unwanted twist or self-intersections into the blending surface. This is illustrated in Figure 5.1. The first surface is an integral B-spline planar patch of the form given in equation (3.20), with knot vectors in the u and v directions of

$$U = (0, 0, 1, 1) \qquad V = (0, 0, 1, 1)$$

and control polygon vertices in three-dimensional space

$$\mathbf{P}_{0,0} = (0, 0, 0) \quad \mathbf{P}_{0,1} = (1, 0, 0) \quad \mathbf{P}_{1,0} = (0, 1, 0) \quad \mathbf{P}_{1,1} = (1, 1, 0)$$

The linkage curve is linear with knot vector and control polygon vertices in the parameter space of the patch

$$X = (0, 0, 1, 1), \quad \mathbf{P}_0 = (0.2, 0.0) \quad \mathbf{P}_1 = (0.7, 1.0)$$

The second surface patch is also planar with the same u, v knot vectors and control polygon vertices of

$$\mathbf{P}_{0,0} = (1, 0, 0) \quad \mathbf{P}_{0,1} = (0, 0, 0) \quad \mathbf{P}_{1,0} = (1, 0, 1) \quad \mathbf{P}_{1,1} = (0, 0, 1)$$

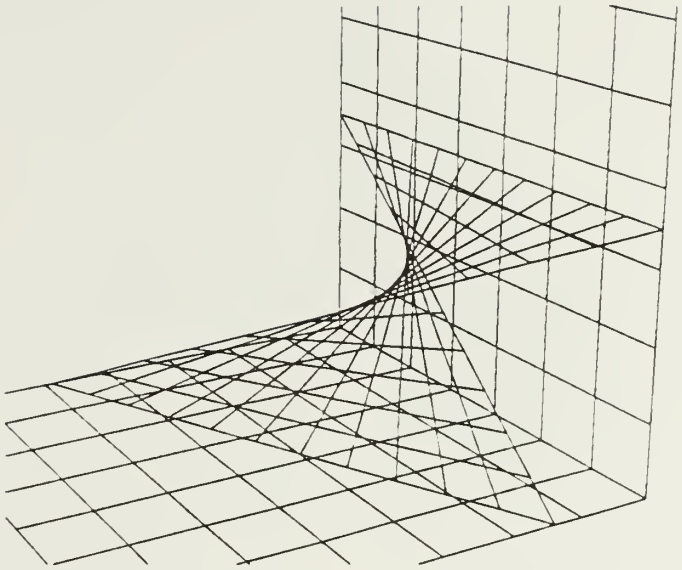


Figure 5.1: Poorly parameterized linkage curves

The linkage curve is linear with knot vector and control polygon vertices in the parameter space of the patch

$$X = (0, 0, 1, 1), \quad \mathbf{P}_0 = (0.5, 0.0) \ \mathbf{P}_1 = (0.5, 1.0)$$

The directional curve is the intersection of the two surfaces expressed in three-dimensional space as a linear spline with knot vector and control polygon vertices

$$X = (0, 0, 1, 1), \quad \mathbf{D}_0 = (0, 0, 0) \ \mathbf{D}_1 = (1, 0, 0)$$

A minimum number of cross-link curves with position continuity boundary conditions along both linkage curves are constructed. Verification of appropriate correspondence of the linkage curve parameterization is a visual check. As shown in Figure 5.1, the linkage curves are parameterized in opposite directions, introducing a twist in the blending surface. To remove the twist, the linkage curve on the second patch

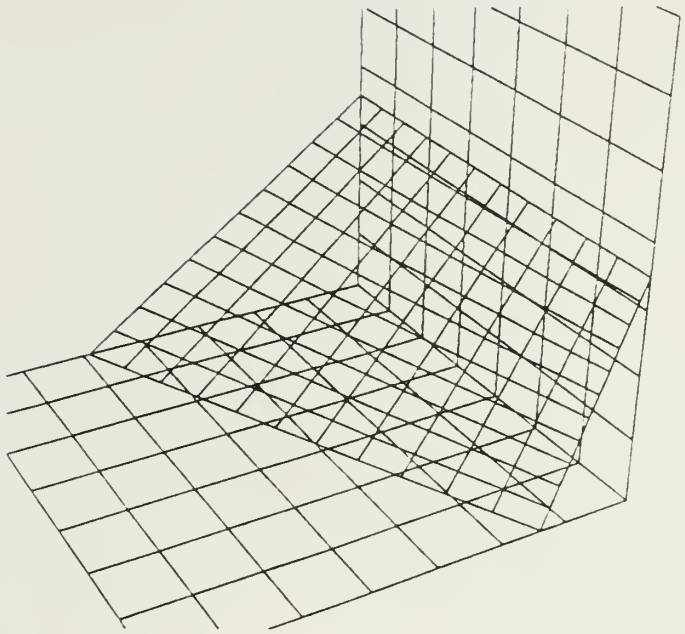


Figure 5.2: Satisfactory parameterization check

is reparameterized with new control points $\mathbf{P}_0 = (0.5, 1.0), \mathbf{P}_1 = (0.5, 0.0)$, and the visual correspondence check performed again. The results of the new parameterization are shown in Figure 5.2.

5.2 Bias Function Effects in Tangent and Cross-Link Curve Curvature Continuous Blends

As noted in Section 4.3.2, the bias function is used to control the shape of the blending surface, and can be given as a constant value or as a function based on the parameterization of the linkage curves or directional curve. The bias function determines the magnitude of the first derivatives of the cross-link curves at their endpoints. A constant value bias function was implemented by [Bardis 89]. The technique requires the distance between corresponding points on the linkage curves to be relatively con-

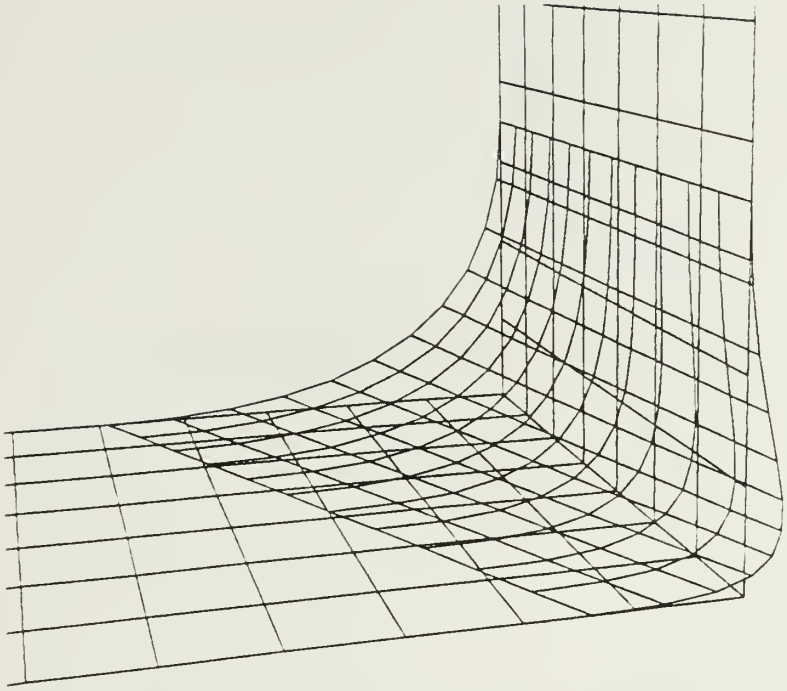


Figure 5.3: Constant scalar bias function, $q_j = 0.4$

stant. As shown in Figure 5.3, if the distance varies significantly, a constant bias factor, suitable where the linkage curves are far apart, can result in self-intersections or cusps in the cross-link curves and the resulting blending surface, or gouging of the underlying surfaces where the linkage curves are closer together. The patches, linkage curves and directional curve are the same as those used in Figure 5.2. The blending surface is tangent plane continuous across both linkage curves, with q_j the scalar bias functions as in equations (4.27) and (4.30), $j = 1, 2$ referring to the corresponding underlying surfaces and linkage curves. An alternative biasing method is a function based on the relative distance between corresponding points on the linkage curves.

$$q_j = \frac{|\mathbf{E}_1(\xi_i) - \mathbf{E}_2(\xi_i)|}{b_j}. \tag{5.1}$$

This formulation automatically accounts for changes in relative linkage curve position, yet allows for user control of the blend shape through variation of the factor, b_j. The

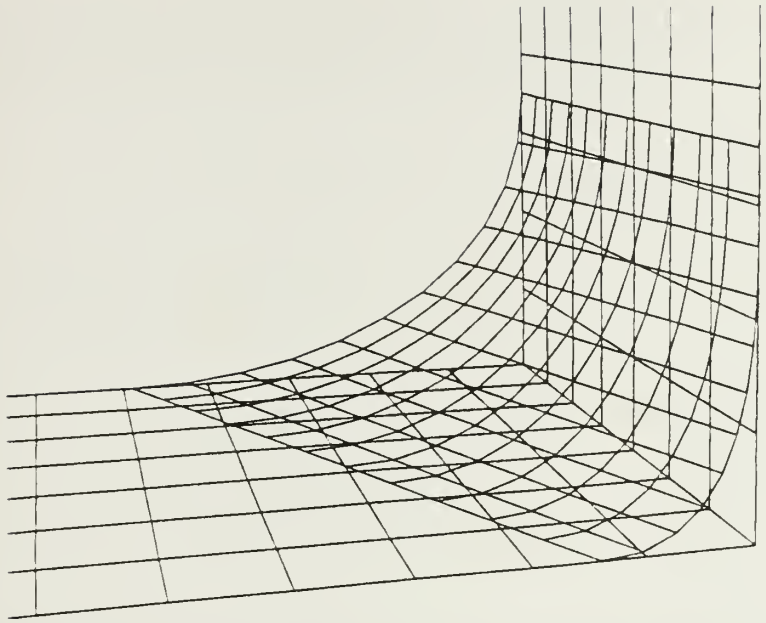


Figure 5.4: Distance function bias factor, $b_j = 2.5$

scalar factors b_j are defined by the user for each linkage curve independently, and can themselves be constant values or functions of the parameterization of the linkage curves. An example of a blending surface using the distance formulation is shown in Figure 5.4. In Figure 5.3, the scalar bias function is $q_1 = q_2 = 0.4$, with user-defined factors b_j specified such that $q_j = constant$. In the distance formulation of Figure 5.4, $b_1 = b_2 = 2.5$ and the bias functions as given by equation (5.1) are now $q_1 = q_2 = 0.4|\mathbf{E}_1(\xi_i) - \mathbf{E}_2(\xi_i)|$. The additional distance term adjusts the magnitude of q_j to prevent the unwanted intersection of the underlying surface shown in Figure 5.3.

Blending operations are local surface operations which must be carefully controlled to ensure the global geometry of the underlying surfaces is not changed. As noted in Chapter 1, the bias function gives the designer a measure of control over the interior of the blending surface. If the magnitudes of the bias factors are not appropriate, the blending surface can gouge one or both of the underlying surfaces. We give four examples of the effect of bias factor magnitudes.

The first surface is a one quarter of a cylindrical surface expressed as a rational Bézier surface patch of orders 3x2. The knot vectors are

$$U = (0, 0, 0, 1, 1, 1) \qquad V = (0, 0, 1, 1)$$

The control polygon vertices are

$$\begin{aligned} \mathbf{P}_{0,0} &= (0, 2, 0, 1) & \mathbf{P}_{0,1} &= (0, 2, 2, 1) \\ \mathbf{P}_{1,0} &= (\sqrt{2}, \sqrt{2}, 0, \frac{1}{\sqrt{2}}) & \mathbf{P}_{1,1} &= (\sqrt{2}, \sqrt{2}, \sqrt{2}, \frac{1}{\sqrt{2}}) \\ \mathbf{P}_{2,0} &= (2, 0, 0, 1) & \mathbf{P}_{2,1} &= (2, 0, 2, 1) \end{aligned}$$

The linkage curve is a linear spline and has the knot vector and control points

$$X = (0, 0, 1, 1) \quad \mathbf{P}_0 = (0.5, 1.0) \quad \mathbf{P}_1 = (0.5, 0.0)$$

expressed in the parameter space of the surface. The second surface is an integral, bilinear planar B-spline surface with knot vectors

$$U = (0, 0, 1, 1) \qquad V = (0, 0, 1, 1)$$

and control polygon vertices

$$\mathbf{P}_{0,0} = (1, 0, 0) \quad \mathbf{P}_{0,1} = (1, 0, 2) \quad \mathbf{P}_{1,0} = (3, 2, 0) \quad \mathbf{P}_{1,1} = (3, 2, 2)$$

The linkage curve is a linear spline expressed in the parameter space of the surface with knot vector and control polygon vertices

$$X = (0, 0, 1, 1) \quad \mathbf{P}_0 = (0.7, 1.0) \quad \mathbf{P}_1 = (0.7, 0.0)$$

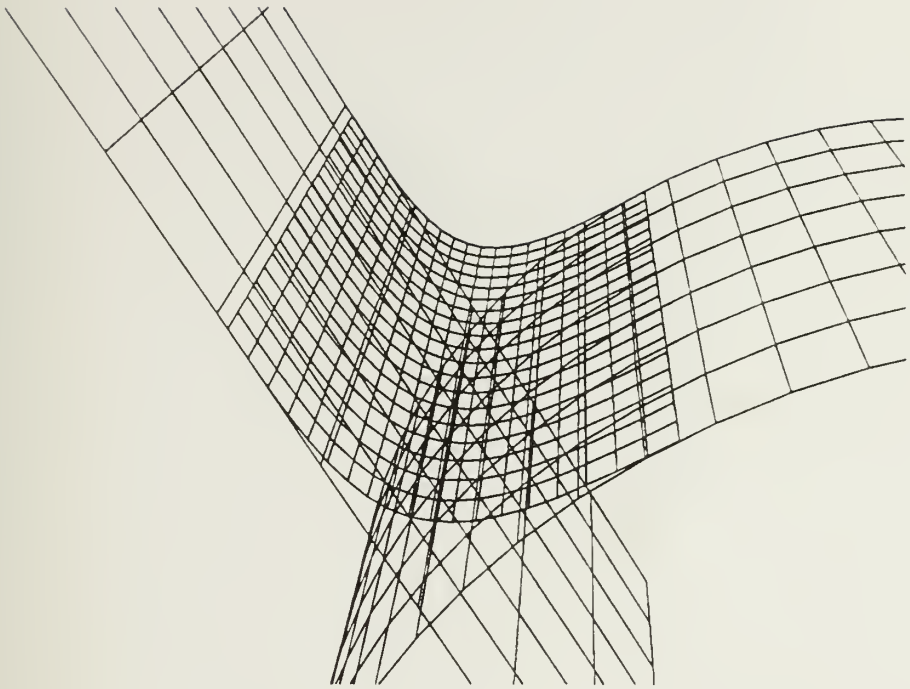


Figure 5.5: Curvature continuous blend, bias factor $b_j = 6$

The directional curve is a linear spline with knot vector and control polygon vertices

$$Y = (0, 0, 1, 1) \quad \mathbf{D}_0 = (2, 0, 2) \quad \mathbf{D}_1 = (3, 2, 2)$$

In the first example we show a well-specified blending surface. The bias factors are equal to $b_j = 6$ along each linkage curve with curvature continuity specified. The blending surface is shown in Figure 5.5. The bias factors are then increased to $b_j = 12$ for each surface. Recalling the distance function as given in Equation 5.1, a larger bias factor weakens the biasing effect. We will call this a weak bias, as compared to a small bias factor which gives a strong bias condition. This is clearly demonstrated in Figure 5.6. For very strong bias factors, especially with curvature continuity specified along the linkage curve, it is visually difficult to tell the difference between the underlying surface and the blend. Furthermore, the stronger the biasing, the further the associated boundary effect is carried into the interior of the blend. If the bias factor is too strong, the blending surface will intersect the other underlying

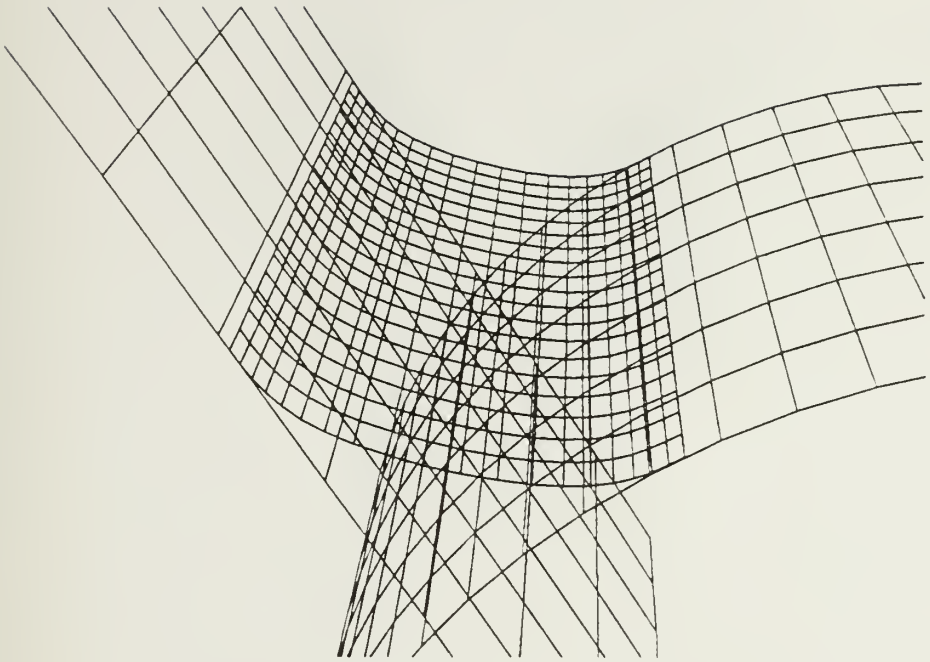


Figure 5.6: Curvature continuous blend, bias factor $b_j = 12$

surface. In Figure 5.7, we try to pull the blending surface towards the planar surface by specifying an imbalance between the bias factors. The bias factor for the plane is $b_2 = 3$, while the bias factor for the cylindrical surface is $b_1 = 10$. The cylindrical surface is gouged by the blend because the biasing at the linkage curve on the plane is too strong. We found that to eliminate this type of gouging, this biasing must be reduced.

Increasing the biasing associated with the gouged surface just exacerbates the problem, creating not only inappropriate intersections of the blending surface with both underlying surfaces but self-intersections of the blending surface as well (see Figure 5.8).

5.3 Marine Propeller Fillets

Modern marine propellers represent particularly complex examples of sculptured surface engineering. Demanding performance criteria require accurate machining and

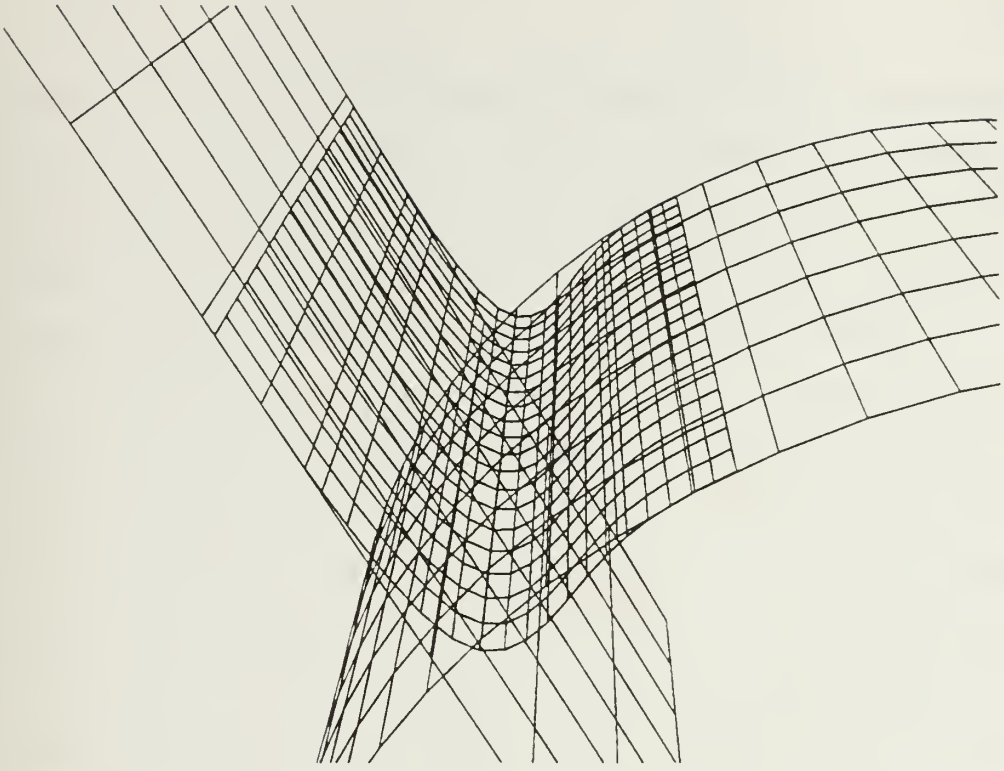


Figure 5.7: Blend with gouging, bias factors $b_1 = 10, b_2 = 3$

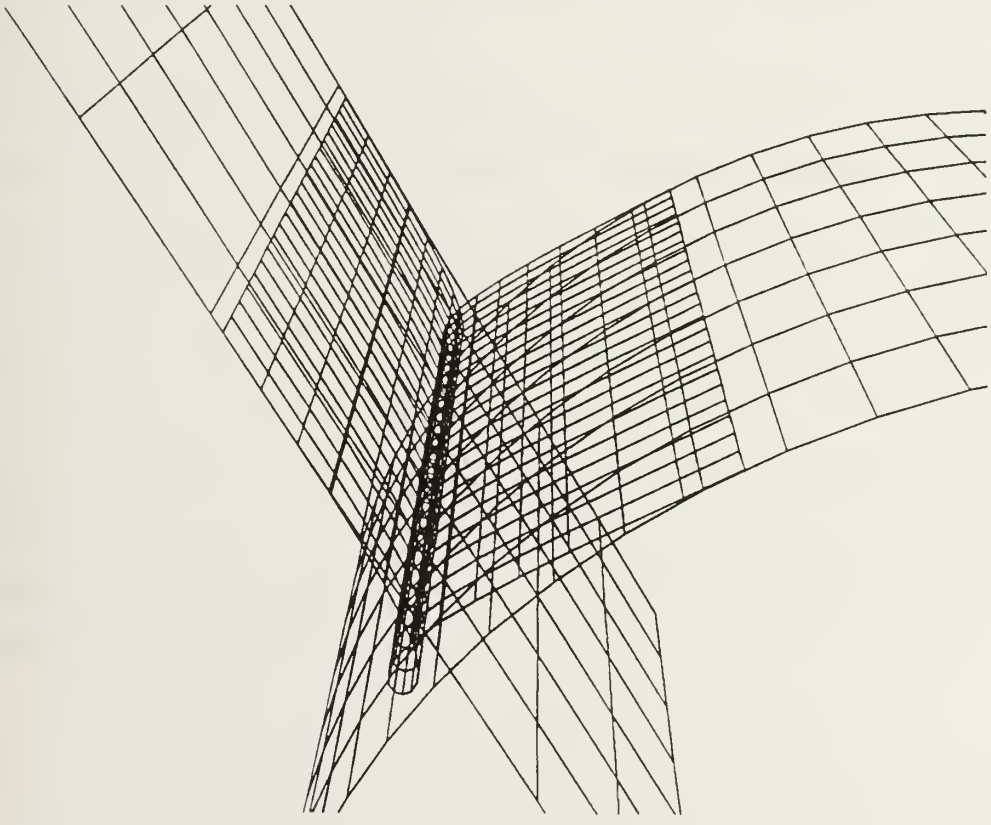


Figure 5.8: Blend with loop, bias factors $b_j = 3$

high surface quality. Hand finishing techniques are usually inadequate. As noted in Chapter 1, the use of numerically controlled machining requires explicit mathematical representation of the entire surface, including the propeller blade-to-hub fillet. Failure to properly represent the fillet may result in removal of fillet material during blade machining, or gouging of the blade faces or hub may occur.

The propeller fillet is both a structural and hydrodynamic surface addressing concerns which arise at the intersection of the propeller blade with the hub. The fillet contributes to the structural strength of the blade and aids in the reduction of vibrational blade loading [Allen 90]. By replacing a sharp edge intersection with a smooth transition surface, adverse hydrodynamic effects are also reduced.

In Chapter 3, it was noted that parametric B-spline surfaces are ideally suited for the representation of complex surfaces. Creation and interrogation of marine propeller surfaces represented with NURBS surfaces is presented in [Tuohy 91c] and [Patrikalakis 90a]. A brief introduction to propeller geometry is provided in Appendix A.

A blending surface for the propeller blade-to-hub fillet will be constructed using two techniques. The first method uses the blending surface algorithm demonstrated earlier in this chapter. The second method is an automatic blending surface designed to replicate a standard propeller fillet geometry called the 3T-T/3 fillet [Gorman 79] [Sabol 83]. A detailed discussion of the geometric requirements and implementation is included in Appendix B. The blending surface resulting from this implementation is a truly rational B-spline surface. For comparison, tangent plane and curvature continuous integral B-spline fillet blends are created using the general algorithm with linkage curves defined by the 3T-T/3 method. These blends are compared with the rational B-spline surface.

5.3.1 Integral B-Spline Surface Fillet

As the first example of the blending surface method applied to the propeller fillet problem, an integral B-spline blending surface is created between a B-spline surface representing a propeller blade and a rational B-spline surface representing a hub. A wire frame of the propeller blade, which has a maximum radius of ten, is shown in Figure 5.9. The hub is a linear by quadratic NURBS surface representing a cylinder of radius one. The linkage curve on the propeller is an isoparametric curve corresponding to $v = 0.2$. The $v = 0$ isoparameter curve of the propeller blade is selected as the directional curve. To find the linkage curve on the hub, the directional curve is projected orthogonally onto the hub surface. The linkage curve is the geodesic offset of the projection, with an offset distance of 0.2 units in three-dimensional space. The linkage curve is expressed in the parameter space of the hub. All of the curves were constructed interactively using *Praxiteles*. Two blending surfaces were created using these surfaces and linkage curves. Figure 5.10 shows a tangent continuous blending surface with a bias factor $b_j = 2.0$ applied along both linkage curves. The position tolerance specified for the blending surface approximation is 0.01 The achieved tangent angle error from equation (4.51) is 0.2 degrees. Figure 5.11 shows the same blending surface with propeller blade and hub.

The blending surface in Figures 5.12 and 5.13 is curvature continuous with a bias factor $b_1 = 7.0$ along the blade linkage curve, and $b_2 = 9.0$ along the hub. To provide an indication of the performance of the blending algorithm, a series of example blending surfaces with these bias factors were created with different accuracy requirements. The results illustrating achieved position, tangent angle (equation 4.51), and relative curvature error (equation 4.56), are shown in Table 5.1.

5.3.2 3T-T/3 Fillet

The most common method for the design of a propeller blade-to-hub fillet is the 3T-T/3 fillet developed at the Philadelphia Naval Shipyard. This design has been

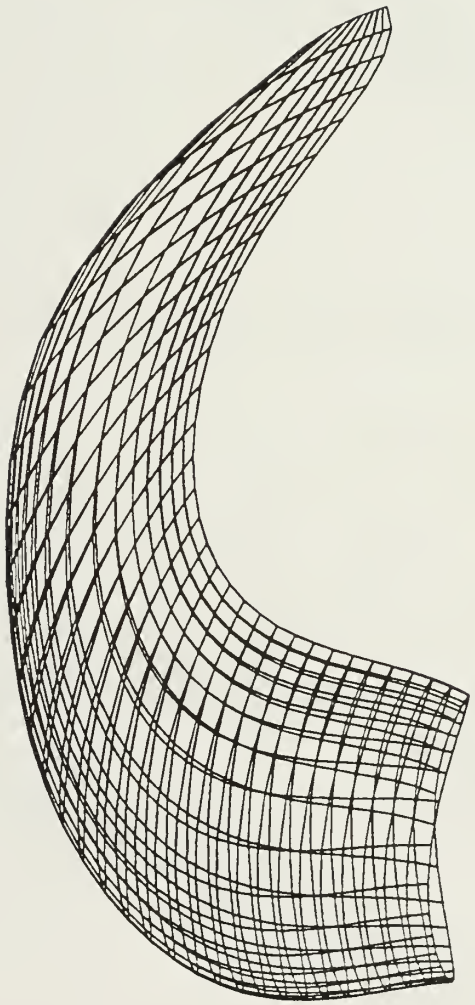


Figure 5.9: B-spline representation of a propeller blade

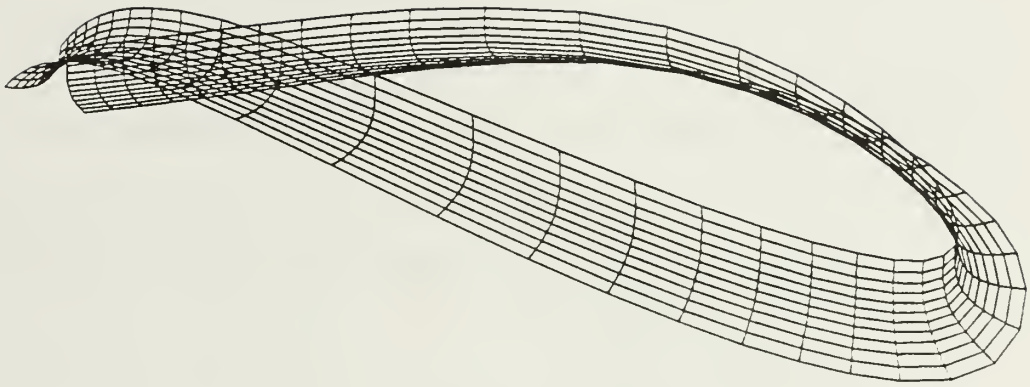


Figure 5.10: Tangent continuous blade-to-hub fillet blend for blade in Figure 5.9

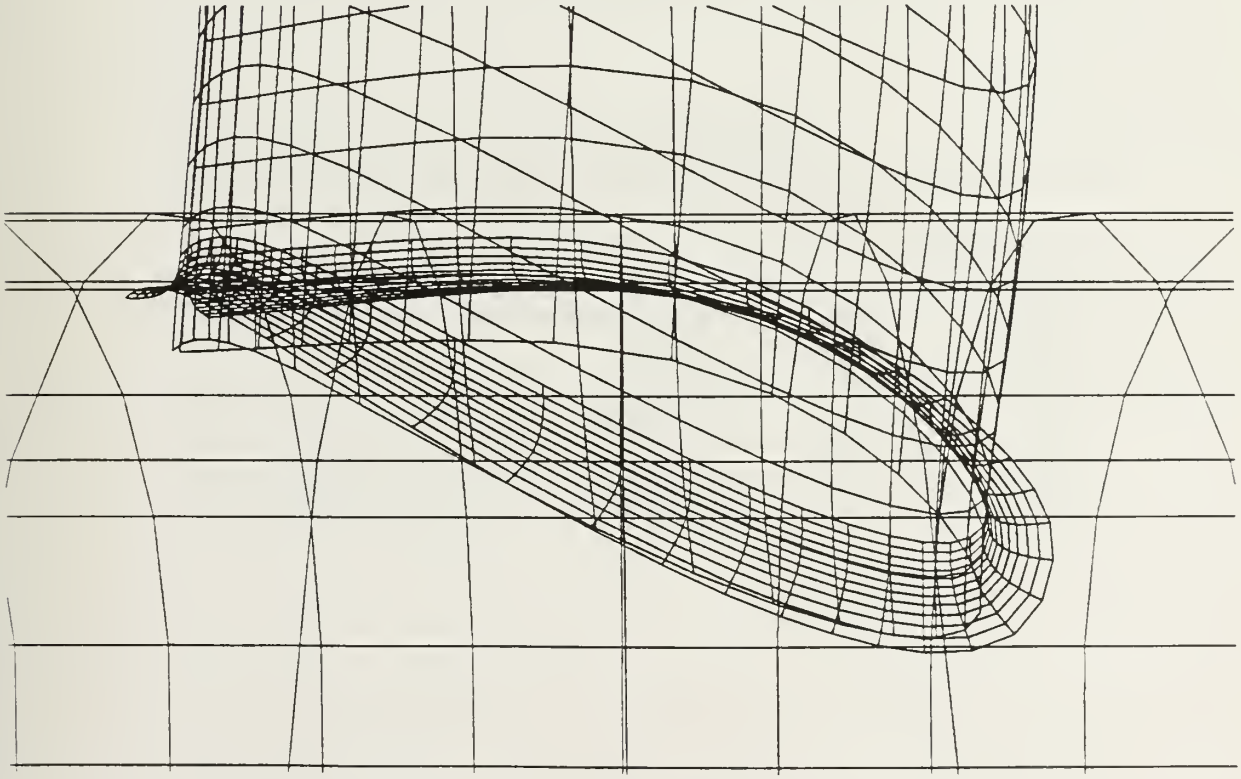


Figure 5.11: Tangent continuous fillet between blade in Figure 5.9 and hub

Table 5.1: Curvature continuous fillet with isoparametric linkage curve of Figure 5.13

Required relative curvature tolerance=10.0% Required tangent angle tolerance=2 degrees					
Required Position Tolerance	Number of Control pts		Achieved Errors		
	u	v	Position	Angle (degrees)	Curvature (Percent)
0.01	41	6	0.00109	0.732	6.40
0.005	53	6	0.000496	0.242	7.23
0.001	73	6	0.000130	0.058	6.61
0.0005	84	6	0.000130	0.058	6.61
0.0001	124	6	0.0000422	0.025	6.44

Required relative curvature tolerance=5.0% Required tangent angle tolerance=2 degrees					
Required Position Tolerance	Number of Control pts		Achieved Errors		
	u	v	Position	Angle (degrees)	Curvature (Percent)
0.01	44	6	0.00109	0.733	4.43
0.005	54	6	0.000496	0.242	4.65
0.001	77	6	0.000130	0.058	5.00
0.0005	88	6	0.000130	0.058	5.00
0.0001	128	6	0.0000422	0.023	4.07

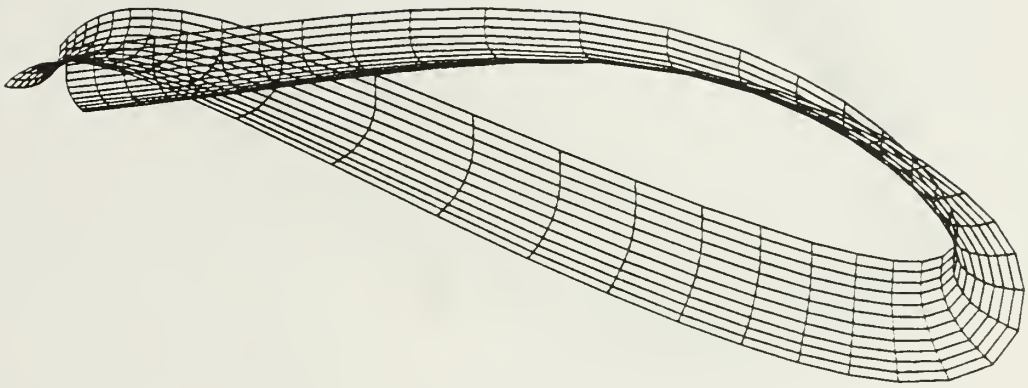


Figure 5.12: Curvature continuous blade-to-hub fillet blend for blade in Figure 5.9

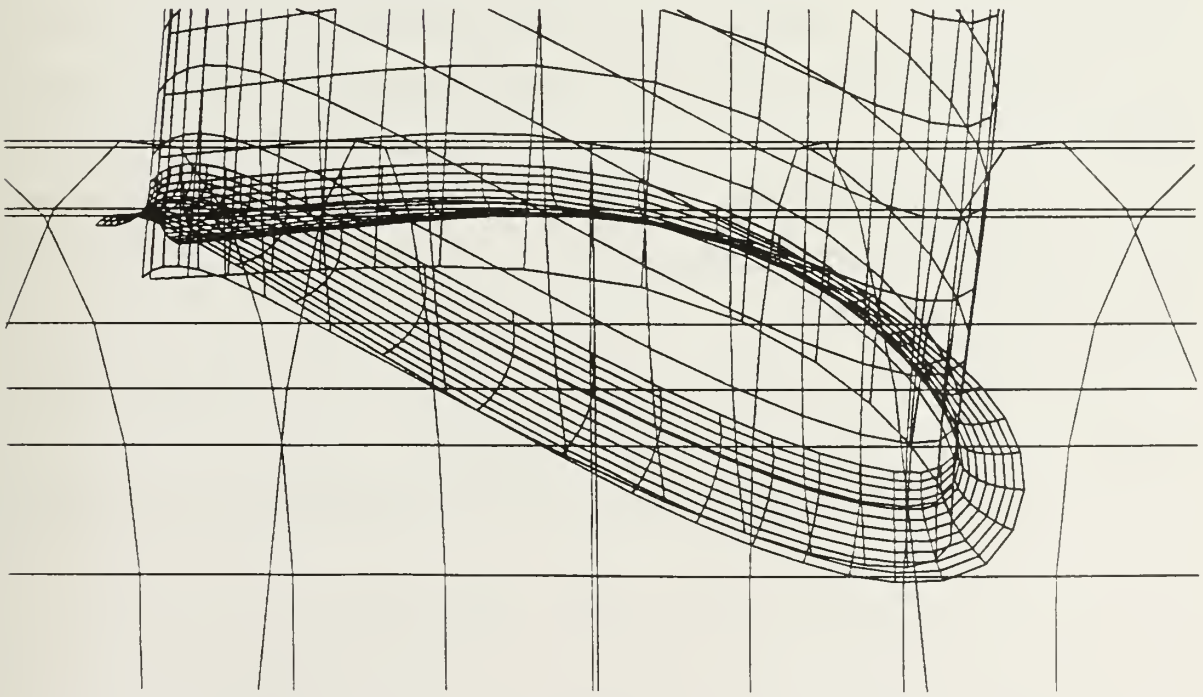


Figure 5.13: Curvature continuous fillet between blade in Figure 5.9 and hub

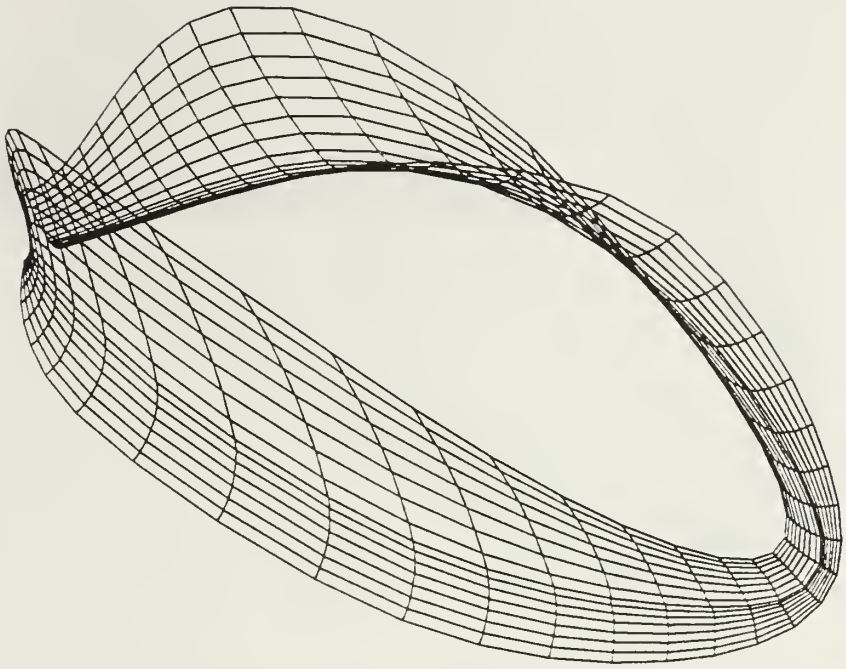


Figure 5.14: 3T-T/3 fillet blend for blade shown in Figure 5.9

adopted by the Naval Sea Systems Command (NAVSEA) as standard geometry for naval propeller fillets [Sabol 83]. Experience has shown that the 3T-T/3 fillet is an effective transition both structurally and hydrodynamically. In Appendix B, the procedure for constructing linkage curves and cross-link curves is presented in detail. The fillet surface is approximated as described in Section 4.4. The fillet is tangent continuous across the linkage curves. Figures 5.14 through 5.16 show the resulting 3T-T/3 fillet surface. The blade in this example is the same as that used in Section 5.3.1. The hub is a linear by quadratic surface of radius 1.25 units. The shape function (Equation B.6) is specified to create a fillet with maximum thickness near mid chord, and 30% of the maximum thickness at the leading and trailing edges. The shape function implemented approximates a graph given in [Sabol 83]:

$$f(x) = 2.1x^2 - 1.4x^3 + 0.3, \qquad 0 \leq x \leq 1$$

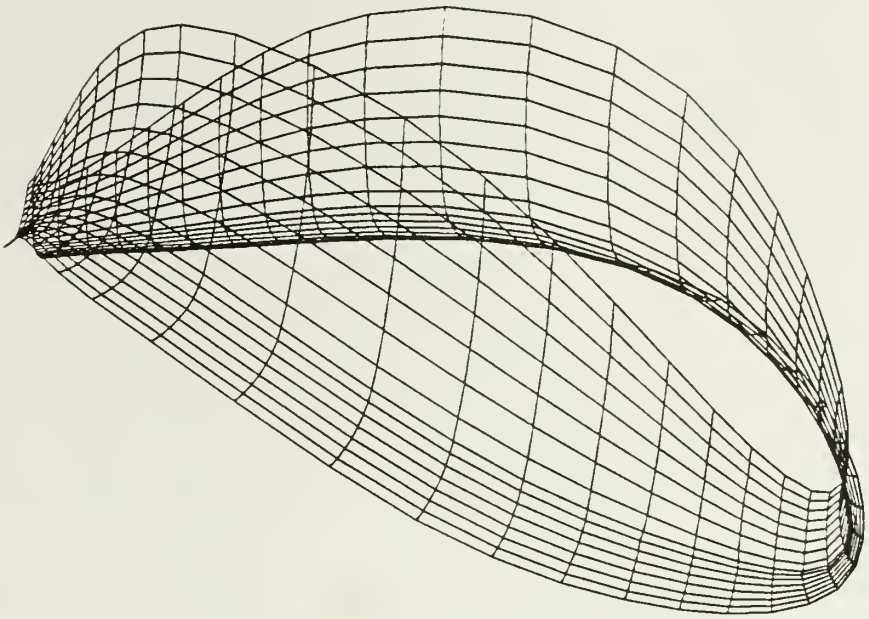


Figure 5.15: 3T-T/3 fillet blend for blade shown in Figure 5.9

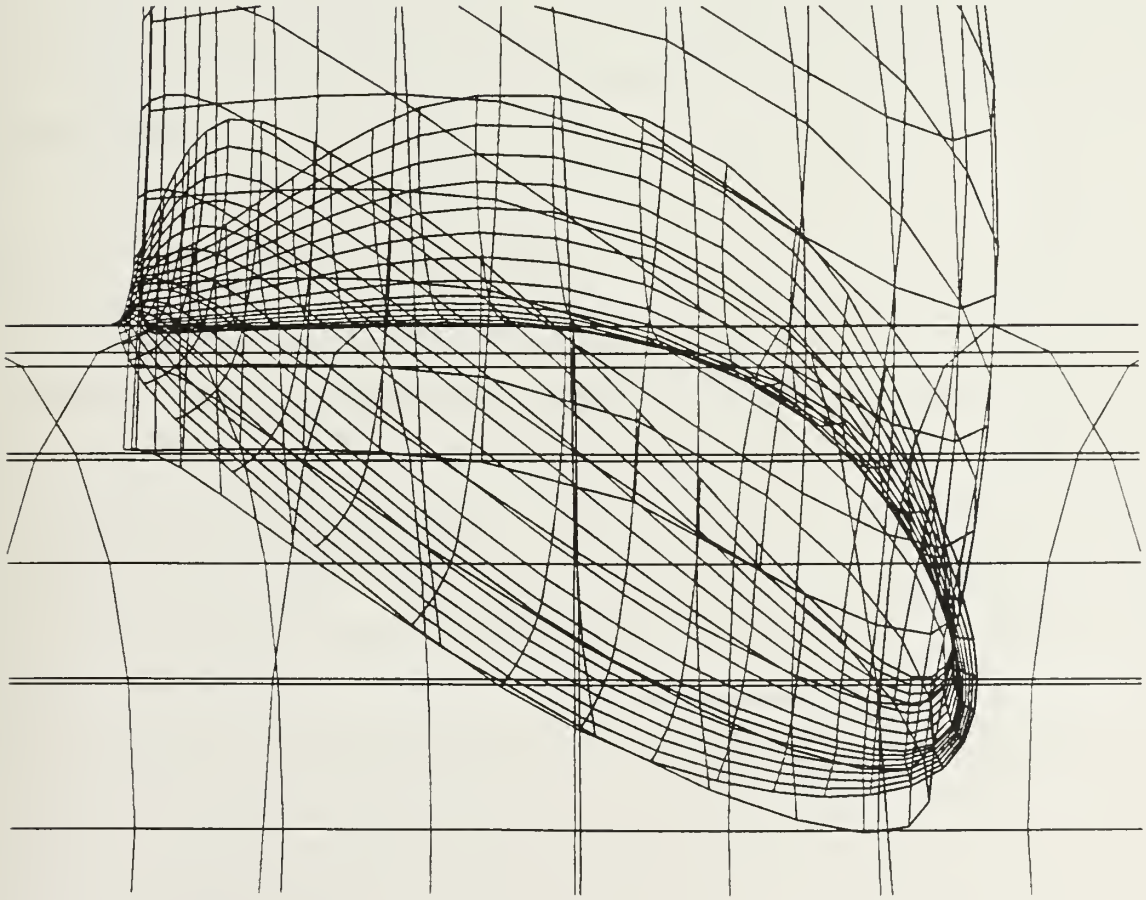


Figure 5.16: 3T-T/3 fillet with blade and hub

The mapping of the parameter u of the intersection curve of the blade to the hub to the parameter x in the shape function $x = x(u)$ is

$$u \in [0, 0.25] \Rightarrow x = 4u \tag{5.2}$$

$$u \in [0.25, 0.5] \Rightarrow x = 4(0.5 - u) \tag{5.3}$$

$$u \in [0.5, 0.75] \Rightarrow x = 4(u - 0.5) \tag{5.4}$$

$$u \in [0.75, 1.0] \Rightarrow x = 4(1 - u) \tag{5.5}$$

The implementation of the 3T-T/3 algorithm is still experimental. As noted in Appendix B, the only error check currently implemented is position continuity of the blending surface with the linkage curve on the hub. For the fillet illustrated in Figure 5.16, the achieved maximum position error is 0.03.

To illustrate the robustness of the algorithm, a 3T-T/3 blend was created using a propeller blade of significantly different shape. The result is shown in Figures 5.17 through 5.19.

5.3.3 Pseudo 3T-T/3 Fillet

The 3T-T/3 fillet was specifically designed to provide a suitable hydrodynamic and structural transition for marine propeller blades. We have no reason to believe the fillet created in Section 5.3.1 has similar properties. For the last two examples, we create propeller fillet blends using the method introduced in Chapter 4, with linkage curves of the 3T-T/3 blend. We will call this blending surface a pseudo 3T-T/3 fillet. In the first example, a tangent plane continuous fillet is created. This blend is most similar to the standard 3T-T/3 fillet since it, too, is tangent plane continuous. The fillet is illustrated in Figures 5.20 and 5.21, using the same propeller blade and hub as was used in Section 5.3.2. A series of examples were created with varying degrees of position accuracy specified. The results are given in Table 5.2. The bias factor b_j for both the blade and the hub is 3.0.

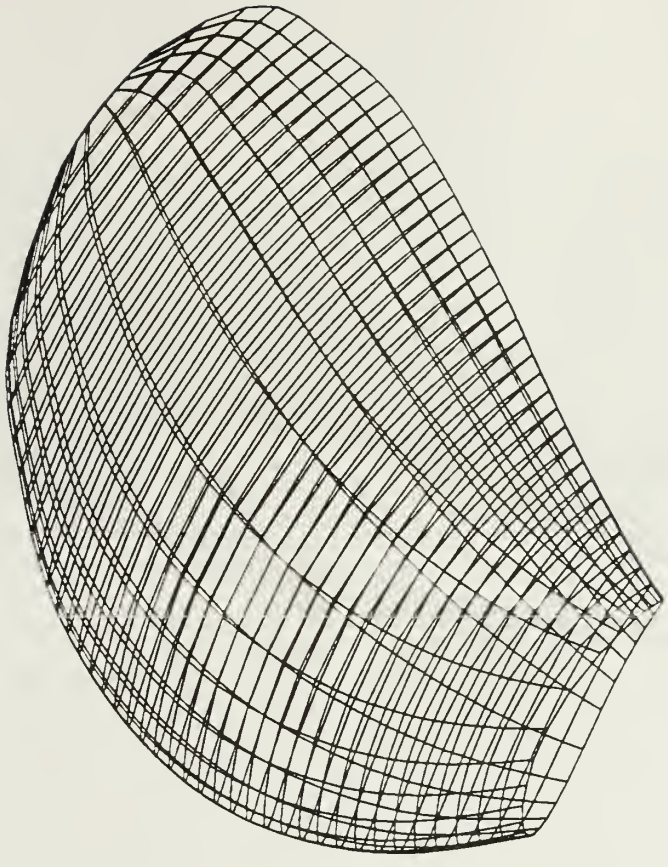


Figure 5.17: Propeller blade

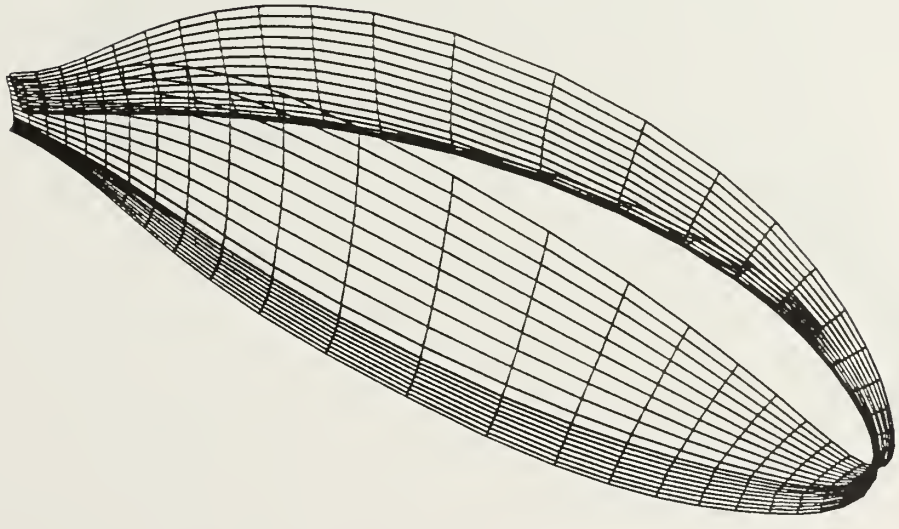


Figure 5.18: 3T-T/3 fillet for blade in Figure 5.17

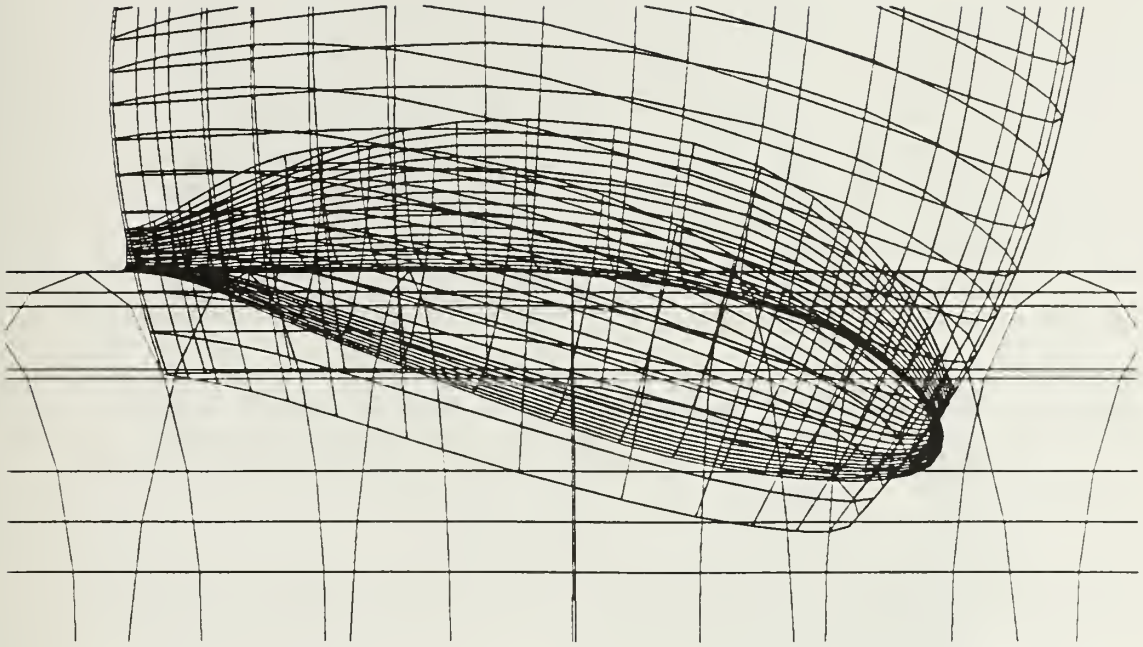


Figure 5.19: 3T-T/3 fillet surface for blade shown in Figure 5.17

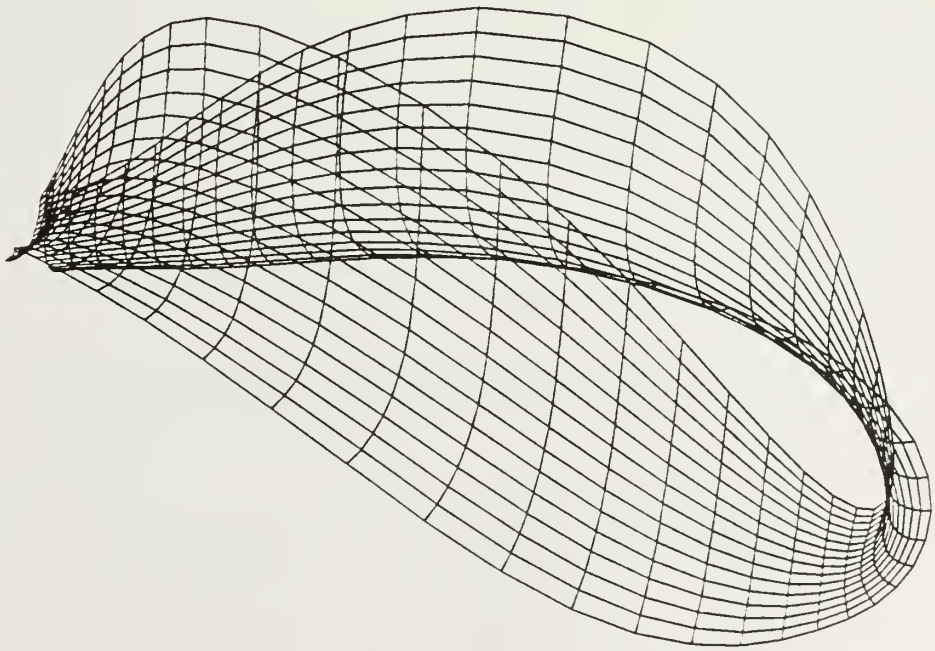


Figure 5.20: Tangent continuous pseudo 3T-T/3 fillet for blade in Figure 5.9

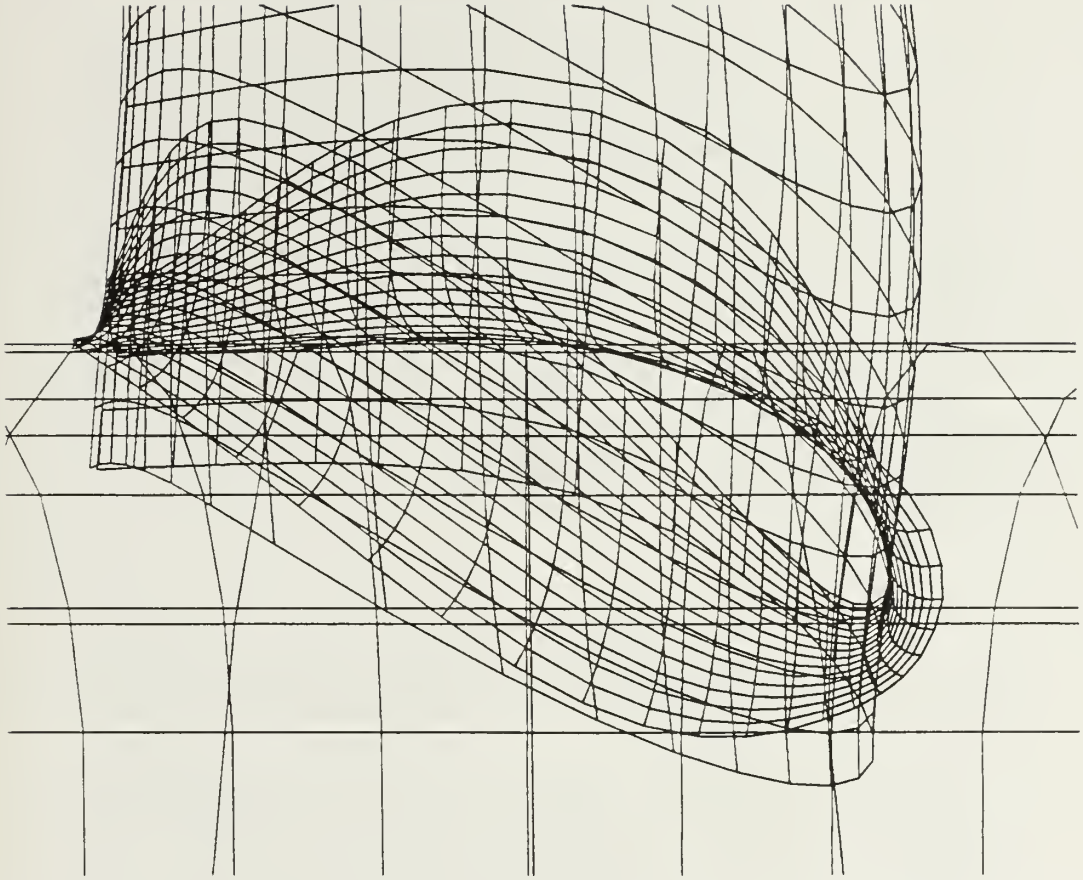


Figure 5.21: Tangent continuous pseudo 3T-T/3 fillet for blade in Figure 5.9

Table 5.2: Tangent continuous pseudo 3T-T/3 fillet

Required tangent angle tolerance=2 degrees				
Required Position Tolerance	Number of Control pts		Achieved Errors	
	u	v	Position	Angle(deg)
0.01	36	4	0.00192	1.35
0.005	44	4	0.000564	0.52
0.001	65	4	0.000161	0.29
0.0005	75	4	0.000114	0.27
0.0001	109	4	0.0000462	0.067

Table 5.3: Curvature continuous pseudo 3T-T/3 fillet

Required relative curvature tolerance=10.0% Required tangent angle tolerance=2 degrees					
Required Position Tolerance	Number of Control pts		Achieved Errors		
	u	v	Position	Angle (degrees)	Curvature (Percent)
0.01	46	6	0.00178	0.602	9.25
0.005	50	6	0.000561	0.517	9.39
0.001	65	6	0.000161	0.290	6.10
0.0005	75	6	0.000114	0.266	4.12
0.0001	109	6	0.0000462	0.0670	5.74
0.00005	128	6	0.0000188	0.0584	2.01

Required relative curvature tolerance=1.0% Required tangent angle tolerance=2 degrees					
Required Position Tolerance	Number of Control pts		Achieved Errors		
	u	v	Position	Angle (degrees)	Curvature (Percent)
0.01	77	6	0.000412	0.0533	0.983
0.005	71	6	0.000556	0.0644	0.901
0.001	79	6	0.000131	0.0581	0.950
0.0005	86	6	0.000114	0.0525	0.956
0.0001	116	6	0.0000462	0.0387	0.765
0.00005	130	6	0.0000188	0.0538	0.801

To complete the examples, a curvature continuous pseudo 3T-T/3 blend is illustrated in Figures 5.22 and 5.23. The bias factor for the blade is $b_1 = 7.0$, and $b_2 = 9.0$ for the hub. These bias factors were selected experimentally to create a blend which most resembles the standard 3T-T/3 fillet. As with the other examples, a series of blending surfaces was created with varying degrees of accuracy. The results are shown in Table 5.3.

Review of the tables indicates the number of control points required to represent a curvature continuous blend is heavily dependent on the specified position tolerance.

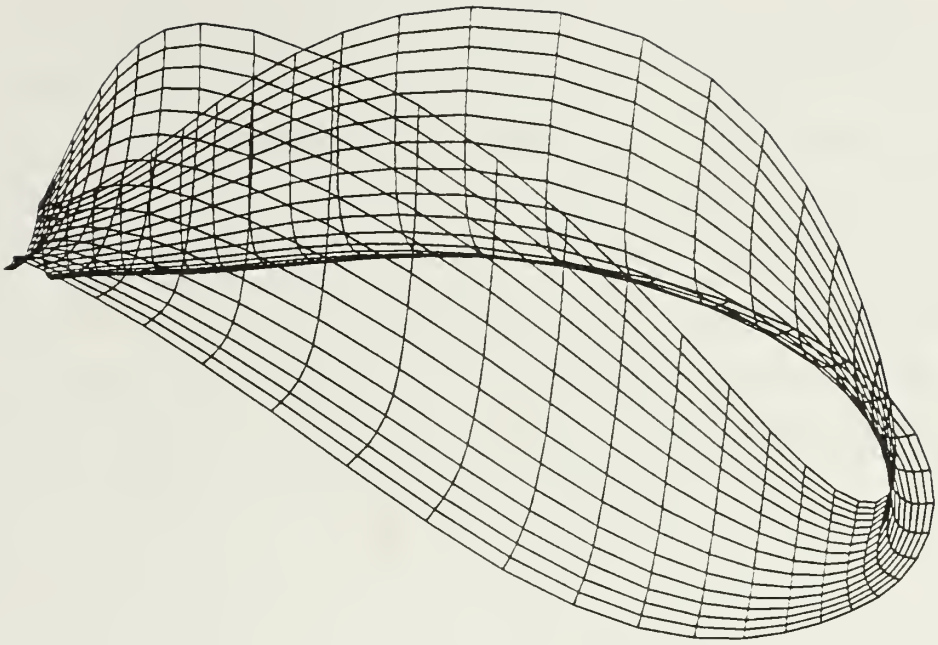


Figure 5.22: Curvature continuous 3T-T/3 fillet for blade shown in Figure 5.9

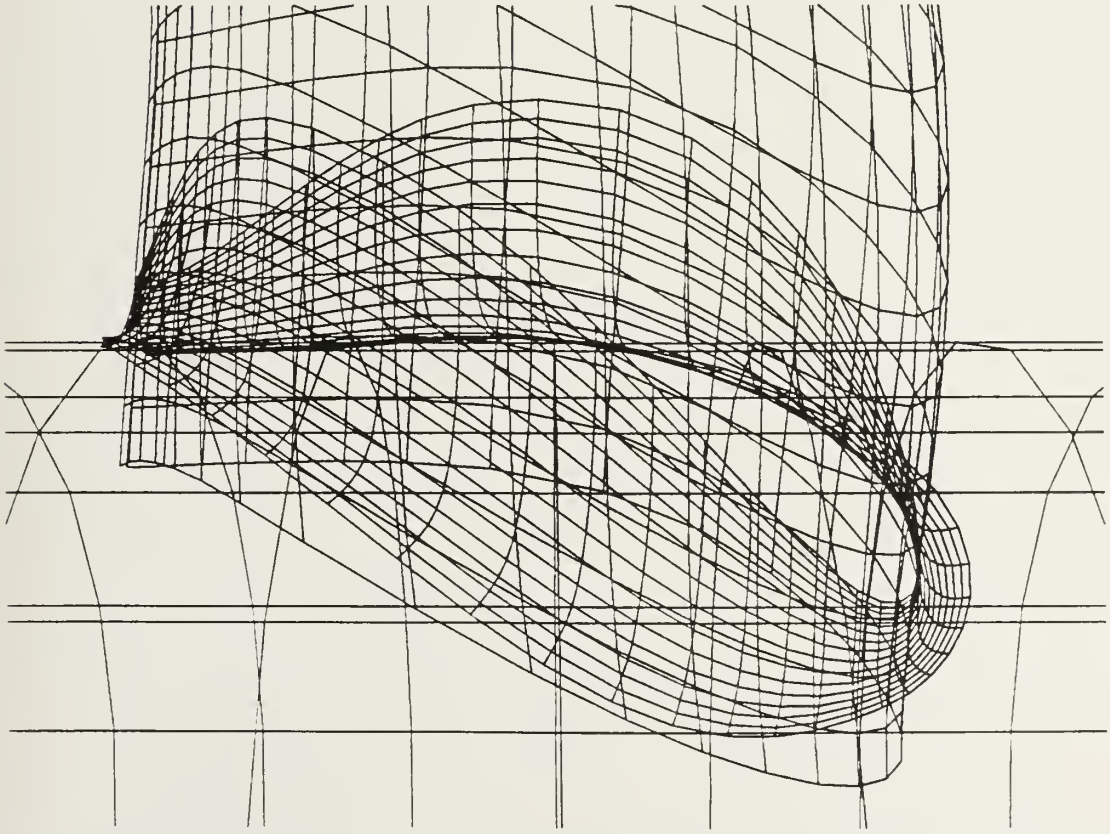


Figure 5.23: Curvature continuous 3T-T/3 fillet for blade shown in Figure 5.9

For a required position tolerance of 0.01, 10 additional control points in the u direction are required to create a blend with a relative curvature error tolerance of 10%, when compared with the tangent continuous blend listed in Table 5.2. Decreasing the relative curvature error tolerance by one order of magnitude for the same position tolerance of 0.01 more than doubles the required number of control points. If the required position tolerance is now reduced to 0.001, the curvature continuous blend with a relative curvature error tolerance of 10% has exactly the same number of control points as the tangent continuous blend. Decreasing the relative curvature error tolerance to 1% requires only 14 additional control points.

Chapter 6

Conclusions and Recommendations

In this thesis, a general algorithm for the representation and approximation of blending surfaces between two non-uniform rational B-spline (NURBS) surfaces using a surface of the same class is presented. As discussed in Chapter 1, blending surfaces are generally procedural surfaces which are difficult to process. Further, there are no standard data exchange formats for procedural surfaces. Our use of NURBS surfaces is important because it facilitates the exchange of geometric information between various geometric modelling systems, and design and manufacturing systems.

The method approximates the blending surface with an integral B-spline surface which is up to curvature continuous with the underlying surfaces along the linkage curves. Several methods are available for specifying the linkage curves. The method allows the degree of continuity along the linkage curves to be specified independently, enhancing its flexibility. Bias functions are control parameters which influence the shape of the blending surface. We demonstrated that if weak biasing is used, the effect of the boundary condition is confined to a narrow region along the linkage curve. Strong biasing propagates the boundary conditions far into the blending surface interior. If the biasing is not carefully controlled, however, inappropriate intersections

with the underlying surfaces can result. Our implementation uses a bias function based on the relative distance between corresponding points on the linkage curves. This technique adjusts the magnitude of the bias so that in regions where the linkage are close together, the probability of gouging the underlying surfaces is reduced but not eliminated. Careful user control is required.

An additional control element is the directional curve. Our implementation guarantees the cross-link curve will be oriented such that its tangent direction at the endpoints is towards the directional curve through the use of an angle comparison algorithm.

The linkage curves are approximated with low degree integral B-spline curves. An adaptive sampling technique adds knots at appropriate locations until the approximation meets all specified tolerances. The cross-link curves are also represented with low degree B-spline curves. The cross-link curves are constructed such that the surface will satisfy the specified continuity conditions at the linkage curves. Methods are implemented for position, tangent plane and curvature continuity along the linkage curves. The cross-link curves then form a family of curves which become the isoparameter curves of a lofted blending surface. The procedure again uses adaptive sampling to improve the approximation. The surface is sampled at the midpoints of the common knot vector formed by merging the knot vectors of the linkage curves. The error checking routine compares the position and surface normal vector of the blending surface with the same information from the corresponding underlying surface. To compare curvature, normal curvature at the endpoints of an isoparameter curve extracted from the blending surface is compared with the normal curvature of a cross-link curve created at the same node using the underlying surfaces in the normal manner. Points which fail to meet the prescribed tolerances are added to the knot vector and the surface relofted. Throughout the improvement process, position accuracy is maintained.

We illustrate the practicality of the method by creating marine propeller blade-to-

hub fillets. Three types of fillets were created with varying degrees of continuity, and each was evaluated over a range of specified position accuracies and, where applicable, relative curvature tolerances. Numerous examples of fillet surfaces with varying degrees of accuracy were examined. We found that required position tolerance with curvature continuity specified has a significant effect on the resulting surface. When the position tolerance is relatively course (of order 0.01), the magnitude of the relative curvature tolerance controls the number of control points necessary to represent the fillet. In the case of the pseudo 3T-T/3 fillet, decreasing the relative curvature tolerance by one order of magnitude nearly doubles the number of required control points. If the position tolerance is smaller, the change in curvature tolerance has less effect. It was demonstrated that position accuracy is maintained during the lofting process, with the achieved position error one to two orders of magnitude better than the position tolerance specified. It was also noted that there is little difference in terms of data storage between representations of tangent continuous fillets and curvature continuous fillets when created with small position tolerances.

The shape of the linkage curves and the underlying surfaces has an impact on the achievable degree of continuity. We found that the fillet designed with an isoparametric linkage curve on the blade could not achieve better than 4% relative curvature error, no matter how small we made the required position tolerance. In comparison, it was possible to create fillets with less than one percent curvature error when using the 3T-T/3 linkage curves, even with a relatively coarse position tolerance of 0.01.

The pseudo 3T-T/3 fillet effectively illustrates the practical application of the algorithm. The rational B-spline 3T-T/3 fillet has the advantage of being automatic in the creation of the fillet. On the other hand, the general algorithm can create a fillet which approximates the rational fillet while retaining the flexibility of user control through the bias functions. Furthermore, the fillet can be either tangent plane or curvature continuous while the rational fillet, by its design, can be at best only tangent plane continuous.

The blending algorithm presented in this thesis has the following advantages:

1. The approximation of the linkage curves and blending surface is adaptive. Sampling and knot addition depends directly on the distribution of error.

2. The blending surface approximation is in a standard format allowing direct exchange between different geometric modelling systems which use the standard data exchange formats [Smith 88]. Further, the surface is of low degree which permits processing on less capable geometric modelling systems.

3. The user has direct control of the shape of the blending surface through specification of boundary conditions, bias functions, and the directional curve.

The following areas are identified for further work:

1. Extend the implementation of the method in Chapter 5 beyond cross-link curve curvature continuity to cover normal section curvature continuity.

2. Provide interrogation methods for the blending surface to ensure there are no unwanted intersections with the underlying surfaces or self-intersections [Kriezis 90]. This might be included as part of the adaptive sampling process.

3. Generalize the algorithm to allow the blending of more than two surfaces.

4. Experimentation with the directional curve to gain greater control of the blend shape. Experimentation with alternate parameterization schemes of the linkage curves and the resulting blending surface.

5. Improving automation of methods for controlling shape and position of the linkage curves. This might include improving the accuracy and efficiency of the rational 3T-T/3 algorithm currently implemented.

6. Automation of bias function specification to eliminate unwanted intersections.

7. Improve methods for visualizing continuity conditions across the boundaries of blending and underlying surfaces.

Appendix A

Marine Propeller Geometry

Marine propeller blade geometry is usually described with reference to a Cartesian coordinate system. The orientation presented here is as suggested by [Kerwin 86]. The x-axis is aligned along the axis of rotation of the propeller and assumed to be positive in the direction of mean flow. The y-axis is vertical and passes through the mid-chord of the root section of one blade. The z-axis is oriented to form a standard right-handed Cartesian coordinate system (see Figure A.1). A cylindrical coordinate system with r the radial coordinate from the x-axis and θ the rotational coordinate, measured about the x-axis such that $\theta = 0$ on the y-axis, is also used.

The propeller blade is typically constructed of a number of two-dimensional foil sections lying on the expanded planes of cylindrical surfaces coaxial with the x-axis. The chord of the section is the two-dimensional straight line segment connecting the leading and trailing edges. In three dimensions the chord is an arc along a helix and is a measure of the pitch angle of the blade at the particular radial distance of the section. The leading edge is determined by advancing, in the negative direction, along a cylindrical or conical helix defined by the radially distributed pitch angle a distance equal to half the chord length. The shape of the foil section is defined in terms of the chordwise distribution of camber, $f(s)$, and thickness, $t(s)$ (see Figure A.2). The parameter s is a coordinate defined along the chord, generally 0 at the leading edge

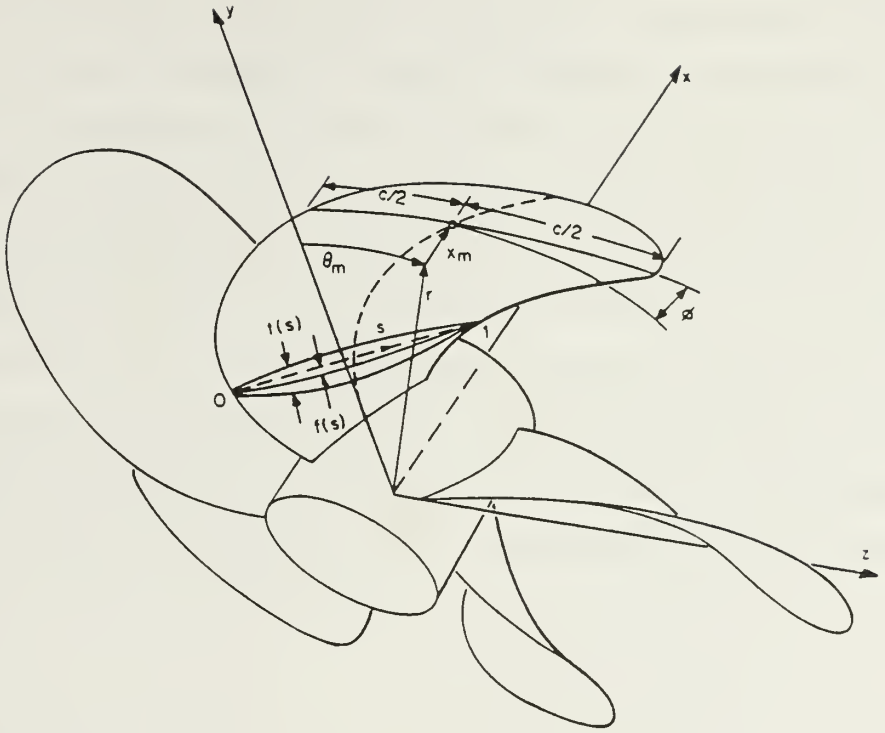


Figure A.1: Propeller blade geometry (from [Kerwin 86])

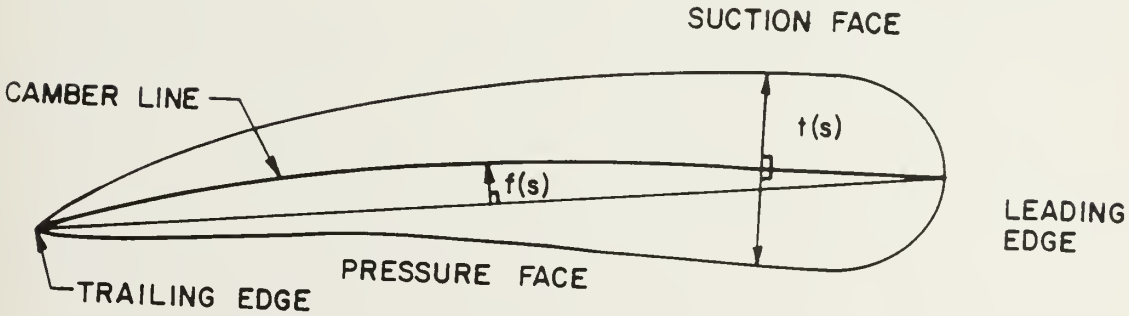


Figure A.2: Foil section

and 1 at the trailing edge. The line traced by $f(s)$ is the camber line, defined as the locus of the midpoints of straight line segments spanned across the section such that the normal to the segment at its midpoint is tangent to the camber line. The camber function defines the shape of the camber line relative to the chord of the section. The half-length of the segments which span the section is defined by the thickness function. These segments define pressure and suction faces of the blade. The pressure face is the blade surface facing aft which experiences an increase in pressure when propelling a ship ahead, while the suction face, or back, experiences a decrease in pressure. The pressure differential results in lift. The root section is the foil section formed by the intersection of the blade with the hub, prior to any blending or filleting operations.

Appendix B

Creation of a 3T-T/3 Marine Propeller Fillet

This Appendix presents a detailed description of our implementation of the standard 3T-T/3 marine propeller fillet using the general algorithm of this thesis.

First, we recall the general method for the creation of a blending surface. We specify linkage curves on the blade surface and the hub. In place of a directional curve we construct the cross-link curves in a fillet plane defined by a geometric function. The cross-link curves are a family of rational B-spline curves which satisfy another specific geometric relationship. Finally, the fillet surface is created by lofting the cross-link curves.

Each cross-link curve is a planar curve constructed of two circular arcs. One of the arcs is tangent to the propeller blade, and one is tangent to the hub surface. In addition, the arcs are tangent to each other. The circular arc tangent to the blade has a radius of 3T, while the arc tangent to the hub has a radius of T/3, hence the name given the fillet. The dimension, T, is defined in section B.1, below. Figure B.1 illustrates the fillet geometry. The generation of the fillet is as follows:

- Compute the intersection curve of the blade and hub, and the maximum thickness of the root section.

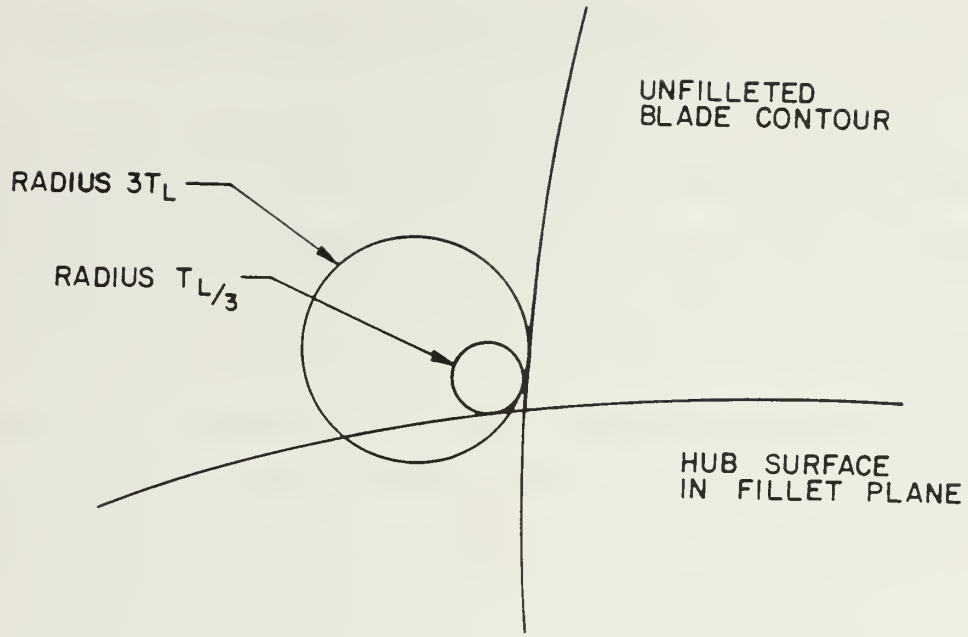


Figure B.1: 3T-T/3 fillet cross-link curve geometry

- Find the linkage curve on the blade surface.
- Establish the fillet plane orientation and the center of the circular arc associated with the linkage curve on the blade.
- Compute the intersection of the fillet plane with the hub.
- Find the center of the T/3 circular arc and its associated contact point on the hub. This is the linkage point on the hub.
- Construct circles in the fillet plane using the computed centers. Subdivide the circles at the points of tangency with the underlying surfaces and with each other.
- Join the arcs to form the cross-link curve.
- Approximate the blending surface from the family of cross-link curves by lofting.
- Sample and improve the approximation.

B.1 Linkage Curve on the Blade

The linkage curve on the blade is defined to lie on the blade surface at an arc length distance T from the blade-to-hub intersection. The original development of the 3T-T/3 fillet suggests that the dimension T is the helical arc length in the root section from the pressure face to the suction face of the unfilleted blade contours. The dimension is a local arc length parameter in that it varies with chord length [Sabol 83]. NAVSEA further recognizes the imprecision of the definition and so provides manufacturing guidance to modify the fillet gradually around the section so that it is fair. [Sabol 83] suggests that T can be defined as a function of the maximum thickness of the root section and a shape factor which varies with chord length.

The maximum thickness of the root section of the blade is found using techniques for feature extraction from B-spline representations of marine propellers presented by [Patrikalakis 90a]. The propeller blade is represented as a bicubic non-uniform integral B-spline surface:

$$\mathbf{R}_{4,4}(u,v) = \sum_{i=0}^{n-1} \sum_{j=0}^{m-1} \mathbf{P}_{i,j} B_{i,4}(u) B_{j,4}(v) \tag{B.1}$$

The parameter u corresponds to the chordwise direction from the trailing edge on the suction face to the leading edge, to the trailing edge on the pressure face. The v parameter corresponds to the spanwise direction, hub to blade tip. The v-isoparameter lines thus approximate the intersections of the blade surface with cylindrical surfaces coaxial with the propeller axis of rotation. Without loss of generality, we assume a cylindrical hub, and the section of interest is the root section formed by the intersection of the cylindrical hub with the blade. Since the intersection curve is known to be a simple regular curve without self-intersections, we use a lattice evaluation technique to find the intersection. Data points representing the intersection of the cylindrical hub surface with isoparameter curves in the v direction (i.e., lines of constant u), extracted from the blade surface hub to tip, are calculated using a simple bisection

method. While more elaborate or efficient methods are available, this one is guaranteed to find the appropriate intersection. The intersection curve $\mathbf{i}(t)$ is approximated as a cubic integral B-spline curve

$$\mathbf{I}(t) = \sum_{i=0}^{n-1} \mathbf{I}_i B_{i,4}(t) \tag{B.2}$$

with knot vector $T = (t_0, t_1, \ldots, t_{n+N-1})$, and $N = 4$. The approximation of the intersection data is similar to the method described in Section 4.2. In this case, (B.2) interpolates the intersection data at the nodes ξ_i of the knot vector

$$\xi_i = \frac{1}{3} \sum_{k=1}^{3} t_{i+k}, \quad 0 \leq i \leq n-1 \tag{B.3}$$

such that

$$\mathbf{I}(\xi_i) = \sum_{j=0}^{n-1} \mathbf{I}_j B_{j,4}(\xi_i) = \mathbf{I}(\xi_i), \quad 0 \leq i \leq n-1 \tag{B.4}$$

Since (B.4) is of the same form as (4.4) it can be written in matrix form,

$$[N]\{\mathbf{I}\} = \{\mathbf{i}\}, \tag{B.5}$$

and, like (4.5), can be solved using standard linear algebra techniques. The approximating B-spline is expressed in terms of normalized cylindrical coordinates. The approximating intersection curve is virtually expanded to a plane by cutting it along a generatrix not intersecting the section. The section curve is thus a planar curve assumed to be at least C^2 continuous, lying on the expanded cylinder. To find the camber line, a correspondence is established between parameters on the pressure and suction faces in accordance with the definition of camber line as given in Appendix A. These are referred to as matching parameters. [Patrikalakis 90a] develops a system of differential equations which, when integrated with respect to the arc length of the section chord, completely define the camber line. The integration proceeds until an irregular point is reached. An irregular point represents a point where the tangent

vectors to matching parameters are parallel to each other. If the segment connecting the matching parameters is also normal to the tangent vectors, the point is called the critical point. [Patrikalakis 90a] shows that there is only one critical point on the camber line and that it corresponds to the maximum thickness of the section. Thus, to find the maximum thickness of the section we use the following procedure:

1. The root section curve is found and approximated as an integral B-spline curve, $\mathbf{I}(t)$.

2. The system of differential equations given in [Patrikalakis 90a] is integrated using a variable step Adams method [NAG 89] until an irregular point is reached.

3. A Newton-Raphson method is then used to accurately solve a nonlinear system of equations to find the critical point. The maximum thickness, T_{max}, of the section is derived directly from the definition of the critical point, knowing the section curve.

Readers interested in the details of the development of the systems of equations are referred to [Patrikalakis 90a].

The parameter T can now be found. As noted earlier, T is a function of the maximum section thickness and a shape factor. The shape factor is defined to produce the maximum fillet thickness near midchord and a minimum thickness near the leading and trailing edges of approximately 1/3 the maximum thickness. In our implementation the shape factor is defined as a general cubic function of the form

$$f(x) = Ax^3 + Bx^2 + Cx + D = 0, \tag{B.6}$$

where $0 \leq x \leq 1$. We define a function $x = x(u)$ (equations (5.2) to (5.5)), to establish a mapping between the variable x and the parameter u of the propeller blade-to-hub intersection curve, expressed in the parameter space of the blade, to achieve the proper shape of the fillet. The local thickness dimension along the intersection curve is then

defined as

$$T_l(u) = T_{max}\, f(x(u)) \tag{B.7}$$

To construct the linkage curve on the blade, we evaluate the approximating B-spline curve of the intersection curve. At each value of u, we extract an isoparameter curve, $u = constant$. The corresponding point on the linkage curve is found along this isoparameter curve, measured in terms of arc length the distance $T_l(u)$. The linkage curve Γ_{prop} is approximated in the parameter space of the propeller blade as before, using as the initial knot vector the knot vector of the approximation to the intersection curve.

The general algorithm requires two linkage curves with each curve defined independently. In the case of the 3T-T/3 blend, however, there is a specific geometric relationship between a point on the linkage curve of the blade and the associated point on the linkage curve of the hub. Because of this dependency, we do not specify the linkage curve on the hub directly but find the corresponding points on the hub while constructing the cross-link curves.

B.2 Fillet Plane

To establish the fillet plane, we evaluate the approximations to the linkage curve on the blade and the blade-to-hub intersection curve at the nodes ξ_i of the common knot vector. At each node there is a unique fillet plane described by the equation [Hildebrand 76]

$$Ax + By + Cz + D = 0. \tag{B.8}$$

The fillet plane is established by the normal vector to the blade surface computed on the intersection curve at a node, and the corresponding point on the linkage curve (see Figure B.2).

The unit normal vector to the blade surface, $\mathbf{n}_{prop}$, at a node of the intersection

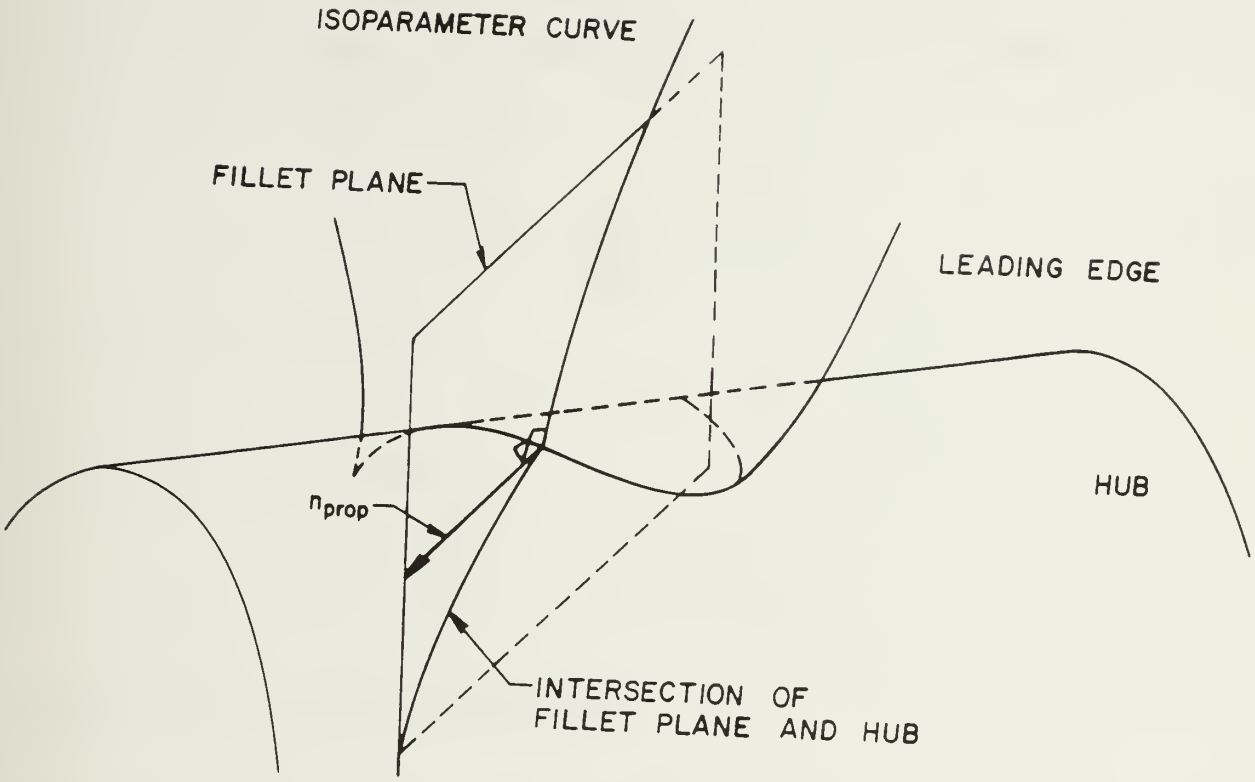


Figure B.2: Fillet plane orientation

curve, is found with equations (4.22) to (4.24). A second vector $\mathbf{p}$ is defined as

$$\mathbf{p}(\xi_i) = \frac{\mathbf{E}_{prop}(\xi_i) - \mathbf{I}(\xi_i)}{|\mathbf{E}_{prop}(\xi_i) - \mathbf{I}(\xi_i)|} \tag{B.9}$$

where $\mathbf{E}_{prop}(\xi_i)$ is the linkage curve position on the blade evaluated at the node ξ_i, and $\mathbf{I}(\xi_i)$ the corresponding point on the approximation to the intersection curve. Let $\mathbf{n}_p$ be a unit vector which is normal to the fillet plane

$$\mathbf{n}_p(\xi_i) = \frac{\mathbf{p}(\xi_i) \wedge \mathbf{n}_{prop}(\xi_i)}{|\mathbf{p}(\xi_i) \wedge \mathbf{n}_{prop}(\xi_i)|}. \tag{B.10}$$

In terms of equation (B.8),

$$\mathbf{n}_p = A\hat{\imath} + B\hat{\jmath} + C\hat{k}. \tag{B.11}$$

The remaining term in (B.8) is

$$D = -\mathbf{n}_{prop}(\xi_i) \cdot \mathbf{E}_{prop}(\xi_i). \tag{B.12}$$

B.3 Cross-Link Curves

To construct the cross-link curves, the centers of the two circular arcs shown in Figure (B.3) and the three points of tangency must be determined.

B.3.1 Centers of Circular Arcs

The center of the 3T circle is found first. The first parametric derivative, $\mathbf{R}^\bullet = \mathbf{R}^\bullet_{4,4}(u, v)$, of the propeller blade in the v direction (3.21) is computed at the corresponding point on the linkage curve, $\mathbf{E}_{prop}(\xi_i)$. The vector $\mathbf{q}$ defined by

$$\mathbf{q} = \mathbf{n}_p \wedge \mathbf{R}^\bullet. \tag{B.13}$$

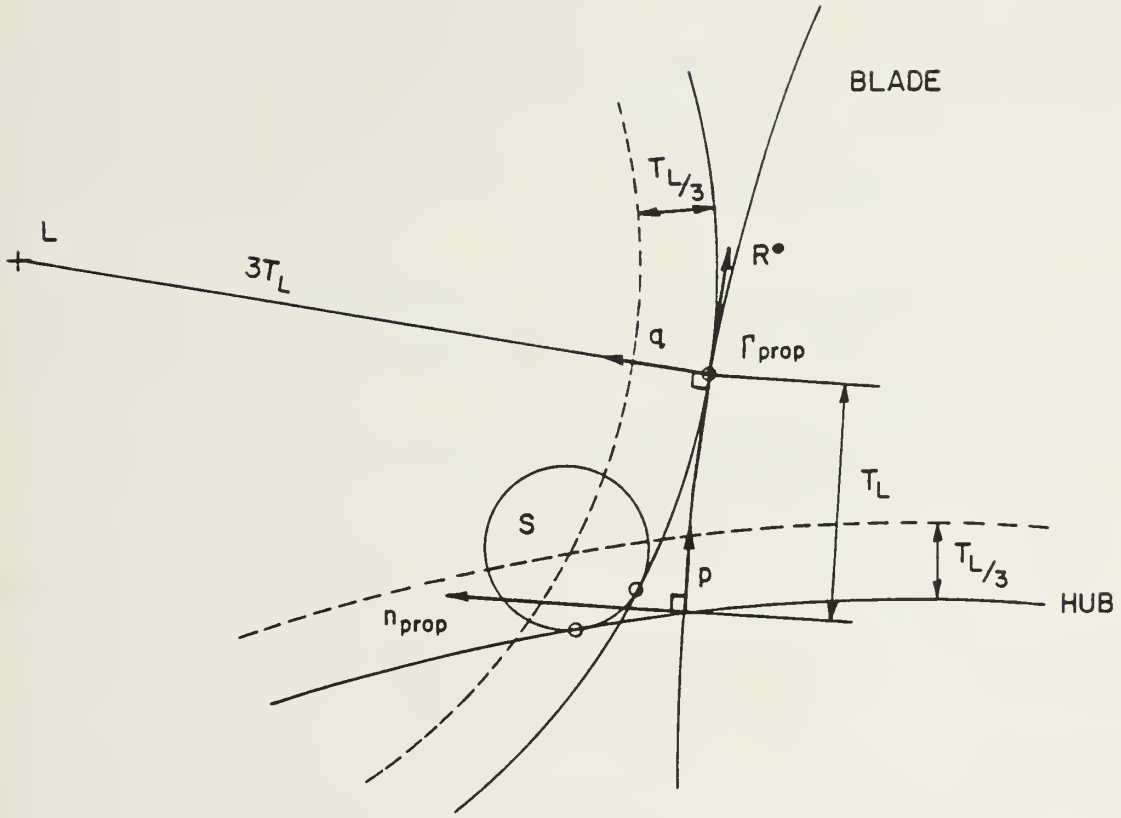


Figure B.3: 3T-T/3 detailed geometry

gives the direction to the center of the 3T circle, **L**.

$$\mathbf{L} = \mathbf{Q}(u, v) + 3T_l \frac{\mathbf{q}}{|\mathbf{q}|} \tag{B.14}$$

where $\mathbf{Q}(u, v)$ is the point on the blade surface corresponding to the point $\mathbf{E}_{prop}(\xi_i)$ on the linkage curve, expressed in three-dimensional coordinates. Likewise, the vector $\mathbf{q}$, defined in Equation (B.13), is evaluated at the corresponding point on the linkage curve.

To locate the center of the T/3 circle, **S**, we must find the intersection of the fillet plane with the hub. The hub is given as a rational B-spline surface,

$$\mathbf{H}_{N,3}(u, v) = \frac{\sum_{i=0}^{n-1} \sum_{j=0}^{8} w_{i,j} \mathbf{P}_{i,j} B_{i,N}(u) B_{j,3}(v)}{\sum_{i=0}^{n-1} \sum_{j=0}^{8} w_{i,j} B_{i,N}(u) B_{j,3}(v)} \tag{B.15}$$

where $\mathbf{H}(u, v) = \langle H_x, H_y, H_z \rangle$, and

$$H_x(u, v) = \frac{\sum_{i=0}^{n-1} \sum_{j=0}^{8} w_{i,j} x_{i,j} B_{i,N}(u) B_{j,3}(v)}{\sum_{i=0}^{n-1} \sum_{j=0}^{8} w_{i,j} B_{i,N}(u) B_{j,3}(v)}. \tag{B.16}$$

H_y and H_z are defined similarly. The intersection of the fillet plane defined by (B.8) is therefore

$$AH_x + BH_y + CH_z + D = 0. \tag{B.17}$$

Since A, B, and C are constants from (B.11), (B.17) can be rewritten as

$$f(u, v) = \sum_{i=0}^{n-1} \sum_{j=0}^{8} w_{i,j}(Ax_{i,j} + By_{i,j} + Cz_{i,j} + D) B_{i,N}(u) B_{j,3}(v) = 0 \tag{B.18}$$

If $f = 0$, then $df = f_u du + f_v dv = 0$, or

$$\frac{du}{dv} = -\frac{f_v}{f_u} \tag{B.19}$$

This formulation requires special treatment when $f_u = 0$ or $f_v = 0$. The difficulty can

be overcome by expressing the intersection curve as a function of the parameter, t, which leads to the following system of differential equations [Kriezis 90]

$$\frac{df}{dt} = f_u \frac{du}{dt} + f_v \frac{dv}{dt} = 0 \tag{B.20}$$

$$\frac{du}{dt} = -\frac{f_v}{\sqrt{f_u^2 + f_v^2}} \tag{B.21}$$

$$\frac{dv}{dt} = \frac{f_u}{\sqrt{f_u^2 + f_v^2}} \tag{B.22}$$

The system of differential equations (B.21) and (B.22) are solved using an Adams-Bashforth technique [NAG 89] to find points along the intersection curve of the fillet plane with the hub surface. The geometric requirements for the T/3 circle are that it

1. be tangent to hub, and
2. be tangent to the 3T circle at a point between the points of tangency of the circles on the hub and blade.

Using simple geometric construction it can be shown that the center of the T/3 circle is located at the intersection of a circle of radius $(3T_l - T_l/3)$, centered at **L**, and a curve in the fillet plane offset from the intersection curve of the fillet plane with the hub at an offset distance $T_l/3$ (see Figure B.4). The center of the circle must be in the fillet plane and, therefore, satisfy (B.8). The second requirement is that it lie on a circle of radius $T_l/3$, centered at **L**. The equation of a sphere, centered at **L** with radius $(3T_l - T_l/3)$ is

$$(x-\xi)^2 + (y-\psi)^2 + (z-\zeta)^2 - (3T_l - \frac{T_l}{3})^2 = 0. \tag{B.23}$$

where (ξ, ψ, ζ) are the three dimensional coordinates of **L**. The intersection of the fillet plane with this sphere is a circle of radius $(T_l - T_l/3)$. We can therefore find the center **S** by restricting the solution to points which satisfy the equation of the fillet plane (B.8). By using the equation for a sphere we free ourselves from working with a

Figure B.4: Center of T/3 circle

local coordinate system. We compute the tangent vector $\mathbf{t}(t)$ to the fillet plane to hub intersection curve by solving the system of differential equations (B.21) and (B.22). The vector $\mathbf{n}^*$ is the vector product of the normal vector to the fillet plane and the tangent vector, $\mathbf{t}$

$$\mathbf{n}^*(t) = \mathbf{n}_p \wedge \mathbf{t}(t). \tag{B.24}$$

The center of the $T_l/3$ circle is found by computing

$$\mathbf{S} = \frac{T_l}{3}\frac{\mathbf{n}^*(t)}{|\mathbf{n}^*(t)|} + \mathbf{H}_{M,N}\left(u(t), v(t)\right) \tag{B.25}$$

using an adaptive step size marching method. The solution of (B.25) which solves both equations (B.8) and (B.23) is the center, $\mathbf{S}$.

B.3.2 Generation of Cross-Link Curves

We now have all the information necessary to create the cross-link curves, namely,

1. the center of the 3T circle, $\mathbf{L} = (L_x, L_y, L_z)$,

2. the point of tangency of the 3T circle with the propeller blade, $\mathbf{Q}(\xi_i) = (Q_x, Q_y, Q_z)$,

3. the center of the T/3 circle, $\mathbf{S} = (S_x, S_y, S_z)$,

4. the point of tangency of the T/3 circle with the hub, and

5. the normal vector to the fillet plane, $\mathbf{n}_p$.

Each circle is defined as a rational B-spline curve

$$\mathbf{A}(t) = \frac{\sum_{i=0}^{8} w_i \mathbf{P}_i B_{i,3}(t)}{\sum_{i=0}^{8} w_i B_{i,3}(t)} \tag{B.26}$$

$$\mathbf{C}(s) = \frac{\sum_{i=0}^{8} h_i \mathbf{K}_i B_{i,3}(s)}{\sum_{i=0}^{8} h_i B_{i,3}(s)} \tag{B.27}$$

where $0 \leq t, s \leq 1$. Let $\mathbf{A}$ be the circle of radius $3T_l$, in the fillet plane with its center at $\mathbf{L}$, and $\mathbf{C}$ the circle of radius $T_l/3$ centered at $\mathbf{S}$, also in the fillet plane. We seek to subdivide these curves at the points of tangency with the underlying surfaces and with each other, then join the segments to form the cross-link curve.

Let the point $\mathbf{Q} = (Q_x, Q_y, Q_z)$ (equation (B.14)) be the coordinates of the point of tangency between $\mathbf{A}(t)$ and the blade surface. We define the parameters t_j, s_j as the values associated with the points of tangency, j, where the circles are to be subdivided. Given $\mathbf{A}(t) = (x(t), y(t), z(t))$, the point of tangency of $\mathbf{A}(t)$ with the blade surface is the value t_1 such that

$$(x(t_1) - Q_x)^2 + (y(t_1) - Q_y)^2 + (z(t_1) - Q_z)^2 = 0 \tag{B.28}$$

is satisfied. A minimization technique [NAG 89] is used to find t_1. A similar expression is evaluated to find the parameter s_1 where $\mathbf{C}$ is tangent to the hub.

We find the parameters associated with the intersection of the two circles in the same manner. In this case, we evaluate the intersection of each circle with a sphere

located at the other center. The circle described by $\mathbf{A}(t)$ lies on the sphere

$$(x(t) - L_x)^2 + (y(t) - L_y)^2 + (z(t) - L_z)^2 - (3T_l)^2 = 0 \qquad \text{(B.29)}$$

Similarly, the smaller circle C lies on the sphere

$$(x(s) - S_x)^2 + (y(s) - S_y)^2 + (z(s) - S_z)^2 - (\frac{T_l}{3})^2 = 0 \qquad \text{(B.30)}$$

At the point of tangency between the two curves, the spheres (B.29) and (B.30) are also tangent. Using a minimization technique with respect to t, we find the intersection of $\mathbf{A}(t)$ with the sphere given by equation (B.30). Similarly, the parameter associated with $\mathbf{C}(s)$ is found minimizing equation (B.29) with respect to s.

Knots are added to the knot vectors of $\mathbf{A}(t)$, $\mathbf{C}(s)$, at the parameter values t_j, s_j, $j = 1, 2$, and the curves subdivided using the Oslo algorithm described in Section 3.2.1. The circular arcs are rational B-spline curves of order 3, geometrically identical to the segments from the original circles between the parameters t_1, t_2 and s_1, s_2, each with a new knot vector normalized over the interval $[0, 1]$. Referring to the segments as $\mathbf{A}(t)$ and $\mathbf{C}(s)$ as before, the curve $\mathbf{A}(t)$ has m control points, while $\mathbf{C}(s)$ has n control points. The segments are now joined to create the cross-link curve.

To join the two curve segments we combine the knot vectors and the control points. This results in a double control point at the intersection of the curve segments, one of which must be removed. Recalling the formulation of a rational B-spline curve,

$$\mathbf{A}(t) = \frac{\sum_{i=0}^{m-1} w_i \mathbf{P}_i B_{i,3}(t)}{\sum_{i=0}^{m-1} w_i B_{i,3}(t)} \qquad \text{(B.31)}$$

$$\mathbf{C}(s) = \frac{\sum_{i=0}^{n-1} h_i \mathbf{K}_i B_{i,3}(s)}{\sum_{i=0}^{n-1} h_i B_{i,3}(s)}. \qquad \text{(B.32)}$$

the double control point is $\mathbf{P}_0$ and $\mathbf{K}_{n-1}$. Because the homogeneous coordinates are

not, in general, equal, we normalize both curves:

$$\mathbf{A}(t) = \frac{\sum_{i=0}^{m-1} w_i h_{n-1} \mathbf{P}_i B_{i,3}(t)}{\sum_{i=0}^{m-1} w_i h_{n-1} B_{i,3}(t)} \tag{B.33}$$

$$\mathbf{C}(s) = \frac{\sum_{i=0}^{n-1} h_i w_0 \mathbf{K}_i B_{i,3}(t)}{\sum_{i=0}^{n-1} h_i w_0 B_{i,3}(t)}. \tag{B.34}$$

where w_0 and h_{n-1} are the homogeneous coordinates of the double control point. We are now free to remove one of the control points at the intersection. The knot vectors are combined as follows. The knot vectors are

$$T = (t_0, t_1, \ldots, t_{m+2}) \tag{B.35}$$

$$S = (s_0, s_1, \ldots, s_{n+2}) \tag{B.36}$$

where T is associated with $\mathbf{A}$, and S with $\mathbf{C}$. To combine the knot vectors, we remove the multiple knots s_n, s_{n+1}, s_{n+2}. Because we remove one control point, the knot t_0 is also removed. Finally, the range of the knot vector T is changed

$$T = (t_1 + 1, t_2 + 1, \ldots, t_{m+2} + 1) \tag{B.37}$$

to keep the knot vectors T and S from overlapping. The combined knot vector spans the interval $[0, 2]$, and is then normalized over the interval $[0, 1]$.

B.4 Approximation of 3T-T/3 Fillet Surface

The approximation of the fillet surface uses the same method as the approximation of the integral B-spline surfaces described in Chapter 4. The knot vectors of the individual cross-link curves are merged to form a common knot vector. The cross-link curves are then lofted to create a rational B-spline surface.

To improve the approximation, we test for position continuity of the blending surface along the boundary of the fillet on the hub. We first find the intersection of the

T/3 circle with the hub as shown in Section B.3.1. for a cross-link curve constructed at the midspan between the nodes of the common vector.

$$t_i = \frac{\xi_i + \xi_{i+1}}{2}. \tag{B.38}$$

The surface approximation error is

$$|\mathbf{S}(t_i, v) - \mathbf{H}(t_i, v)| \quad 0 \leq v \leq 1, \tag{B.39}$$

where $\mathbf{H}(t_i, v)$ is the intersection point of the T/3 circle on the hub expressed in three dimensional space. and $\mathbf{S}(t_i, v)$ is the corresponding point on the blending surface. If the error function is greater than the specified tolerance. ϵ. the point t_i is added to the knot vector. The blending surface is relofted with the enriched knot vector until the blending surface passes the error test at all points t_i.

Bibliography

[Allen 90] D. W. Allen, W. S. Vinoski, and B. A. Overton. Computer aided design methods for propeller fillet gages. *Naval Engineers Journal.* 102(3):202–208, May 1990.

[Alourdas 89] P. G. Alourdas. Shape creation, interrogation and fairing using B-splines. Engineer's Thesis, Massachusetts Institute of Technology, Department of Ocean Engineering, Cambridge, Massachusetts. 1989.

[Alourdas 90] P. G. Alourdas, G. R. Hottel, and S. T. Tuohy. A design and interrogation system for modeling with rational splines. In S. K. Chakrabarti et al., editor, *Proceedings of the Ninth International Symposium on Offshore Mechanics and Arctic Engineering. Houston, Texas, February, 1990*, volume I. Part B, pages 555–565. NY: ASME, 1990.

[Bardis 89] L. Bardis and N. M. Patrikalakis. Blending rational B-spline surfaces. *Eurographics '89, European Computer Graphics Conference and Exhibition, Hamburg, F. R. of Germany, September 1989. pp. 453-462. Amsterdam: North-Holland.*, 1989.

[Bloor 89a] M. I. G. Bloor and M. J. Wilson. Generating blend surfaces using partial differential equations. *Computer Aided Design*, 21:165–171. 1989.

[Bloor 89b] M. I. G. Bloor and M. J. Wilson. Blend design as a boundary-value problem. In *Geometric Modeling: Theory and Practice*, pages 221–234. Springer-Verlag, 1989.

[Bloor 89c] M. I. G. Bloor and M. J. Wilson. Generating n-sided patches with partial differential equations. In *New Advances in Computer Graphics. Proceedings of CG International '89*, pages 129–145. Springer-Verlag, 1989.

[Bloor 90] M. I. G. Bloor and M. J. Wilson. Using partial differential equations to generate free-form surfaces. *Computer Aided Design*, 22:202–212, 1990.

[Celniker 89] G. Celniker and D. Gossard. Energy-based models for free-form surface shape design. *Advances in Design Automation - 1989. NY: ASME*, 1:107–112, 1989.

[Cohen 80] E. Cohen, T. Lyche, and R. F. Riesenfeld. Discrete B-splines and subdivision techniques in computer-aided geometric design and computer graphics. *Computer Graphics and Image Processing*, 14:87–111, 1980.

[Cohen 85] E. Cohen, T. Lyche, and L. L. Schumaker. Algorithms for degree-raising of splines. *ACM Transactions on Graphics*, 4(3):171–181, July 1985.

[Cox 72] M. G. Cox. The numerical evaluation of B-splines. *Journal of the Institute for Mathematics Applications*, 10:134–149, 1972.

[de Boor 72] C. de Boor. On calculating with B-splines. *Journal of Approximation Theory*, 6:50–62, 1972.

[de Boor 78] C. de Boor. *A Practical Guide to Splines.* Springer, New York. 1978.

[do Carmo 76] P. M. do Carmo. *Differential Geometry of Curves and Surfaces.* Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1976.

[Davis 70] R. F. Davis. Marine propulsor blade lofting. Tech. Memorandum File No. TM512-03, Applied Research Laboratory, The Pennsylvania State University, August 1970.

[Farin 88] G. Farin. *Curves and Surfaces for Computer Aided Geometric Design - A Practical Guide.* Academic Press. 1988.

[Faux 79] I. D. Faux and M. J. Pratt. *Computational Geometry for Design and Manufacture.* Ellis Horwood, Chichester, England, 1979.

[Fox 87] C. Fox. *An Introduction to the Calculus of Variations.* Dover Publications, 1987.

[Gorman 79] R. W. Gorman. General mathematical definition of propeller blade-fillet geometry. Tech. Report DTNSRDC-79/053. David Taylor Naval Ship Research and Development Center, August 1979.

[Hansmann 85] W. Hansmann. *Interaktiver Entwurf und geometrische Beschreibung glatter übergänge zwischen räumlich gekrümmten Flächenstrukturen.* PhD thesis, Fachbereich Informatik der Univerität Hamburg, 1985.

[Hansmann 87] W. Hansmann. Interactive design and geometric description of smooth transitions between curved surfaces. In *Computers in Offshore and Arctic Engineering, 6th International Symposium*

on Offshore Mechanics and Arctic Engineering, Houston, Texas.. ASME, NY, pages 19–26, New York, 1987. ASME.

[Hartley 80] P. J. Hartley and C. J. Judd. Parametrization and shape of B-spline curves for CAD. *Computer Aided Design*, 12(5):235–238, September 1980.

[Hildebrand 76] F. B. Hildebrand. *Advanced Calculus for Applications*. Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1976.

[Hoelzle 83] J. E. Hoelzle. Knot placement for piecewise polynomial approximation of curves. *Computer Aided Design*, 15(5):296–296, September 1983.

[Hoffmann 87c] J. Hopcroft and C. Hoffmann. The potential method for blending surfaces and corners. In *Geometric Modeling: Algorithms and New Trends*, pages 347–365. SIAM, 1987.

[Hoschek 87] J. Hoschek. Approximate conversion of spline curves. *Computer Aided Geometric Design*, 4:171–181, 1987.

[Hottel 89] G. R. Hottel. An executive system for modeling with rational B-splines. Engineer's Thesis, Massachusetts Institute of Technology, Department of Ocean Engineering, Cambridge, Massachusetts, 1989.

[Hottel 91] G. R. Hottel, S. T. Tuohy, P. G. Alourdas, and N. M. Patrikalakis. Praxiteles: A geometric modeling and interrogation system. *Cambridge, Mass: MIT Ocean Engineering Design Laboratory Memorandum 91-6*, April 1991.

[Kerwin 86] J. E. Kerwin. *Marine Propellers*, volume 18, pages 307–403. Annual Reviews Inc., Palo Alto, CA, USA, 1986.

[Koparkar 91] P. Koparkar. Designing parametric blends: surface model and geometric correspondence. *The Visual Computer*, 7:39–58, 1990.

[Kriezis 90] G. A. Kriezis. *Algorithms for Rational Spline Surface Intersections.* PhD thesis, Massachusetts Institute of Technology, Cambridge, Massachusetts, March 1990.

[Mortenson 85] M. E. Mortenson. *Geometric Modeling.* John Wiley & Sons, New York, 1985.

[Nittel 89] M. F. Nittel. Numerically controlled machining of propeller blades. *Marine Technology*, 26(3):202–209, July 1989.

[NAG 89] Numerical Algorithms Group, Oxford, England. *NAG Fortran Library Manual*, Mark 13 edition, 1989.

[Patrikalakis 88d] N. M. Patrikalakis and G. A. Kriezis. Representation of piecewise continuous algebraic surfaces in terms of B-splines. *The Visual Computer*, 5(6):360–374, 1989.

[Patrikalakis 89a] N. M. Patrikalakis, L. Bardis, and G. A. Kriezis. Approximation of B-spline geometries with lower degree representations. In *Journal of Offshore Mechanics and Arctic Engineering, ASME Trans., Volume 112, No. 3, 1990, pages 192-198. Also: 8th International Symposium on Offshore Mechanics and Arctic Engineering, Hague, Netherlands., Volume VI, Computer Technology, March 1989, 237-246, NY: ASME.*

[Patrikalakis 89b] N. M. Patrikalakis and L. Bardis. Offsets of curves on rational B-spline surfaces. *Engineering with Computers*, 5:39–46, 1989.

[Patrikalakis 90a] N. M. Patrikalakis and L. Bardis. Feature extraction from B-spline marine propeller representations. Cambridge, MA: MIT Ocean

Engineering Design Laboratory Memorandum 90-12, September 1990.

[Pegna 87] J. Pegna. *Variable Sweep Geometric Modeling.* PhD thesis, Stanford University, Stanford, CA, 1987.

[Pegna 88] J. Pegna and D. J. Wilde. Spherical and circular blending of functional surfaces. In *Proceedings of the 7th International Conference on Offshore Mechanics and Arctic Engineering, Houston, Texas.*, volume 7, pages 73–82. ASME, New York, February 1988.

[Pegna 89] J. Pegna. Interactive design of curvature continuous fairing surfaces. In *Proceeding of the 8th International Conference on Offshore Mechanics and Arctic Engineering, The Hague, The Netherlands.*, volume VI, pages 191–198. ASME, New York, March 1989.

[Pegna 89a] J. Pegna and F. E. Wolter. Geometrical criteria to guarantee second order smoothness of blend surfaces. In *Proceedings of the 15th Design Automation Conference, Montreal, Canada*, volume I, pages 93–105. ASME, New York, September 1989.

[Pegna 90] J. Pegna and F. E. Wolter. Designing and mapping trimming curves on surfaces using orthogonal projection. In *Proceedings of the 16th ASME Design Automation Conference: Advances in Design Automation, Computer Aided and Computational Design. Ravani, B. ed., Chicago, IL, September 1990, Vol. I*, pages 235–245. NY: ASME, 1990.

[Rogers 76] D. F. Rogers and J. A. Adams. *Mathematical Elements for Computer Graphics.* Mc Graw-Hill Inc., 1976.

[Rossignac 84] J. Rossignac and A. Requicha. Constant-radius blending in solid modeling. *Computers in Mechanical Engineering*, pages 65–73, July 1984.

[Rossignac 85] J. R. Rossignac. *Blending and Offseting Solid Models.* PhD thesis, University of Rochester, July 1985. Production Automation Project Technical Memorandum No. 54.

[Sabol 83] W. J. Sabol. Analytic fillet development for marine propulsors. Tech. Memorandum File No. 83-01, Applied Research Laboratory, The Pennsylvania State University, January 1983.

[Sanglikar 90] M. A. Sanglikar, P. Koparkar, and V. N. Joshi. Modelling rolling ball blends for computer aided geometric design. *The Visual Computer*, 7:399–414, August 1990.

[Smith 88] B. Smith, G. R. Rinaudot, K. A. Reed, and T. Wright. *Initial Graphics Exchange Specification (IGES), Version 4.0.* National Bureau of Standards NBSIR 88-3813, 1988.

[Tiller 83] W. Tiller. Rational B-splines for curve and surface representation. *IEEE Computer Graphics and Applications*, 3(6):61–69, September 1983.

[Tuohy 91b] S. T. Tuohy. Sculptured shape creation, approximation, and interrogation. Engineer's Thesis, Massachusetts Institute of Technology, Department of Ocean Engineering, Cambridge, Massachusetts, 1991. (To appear).

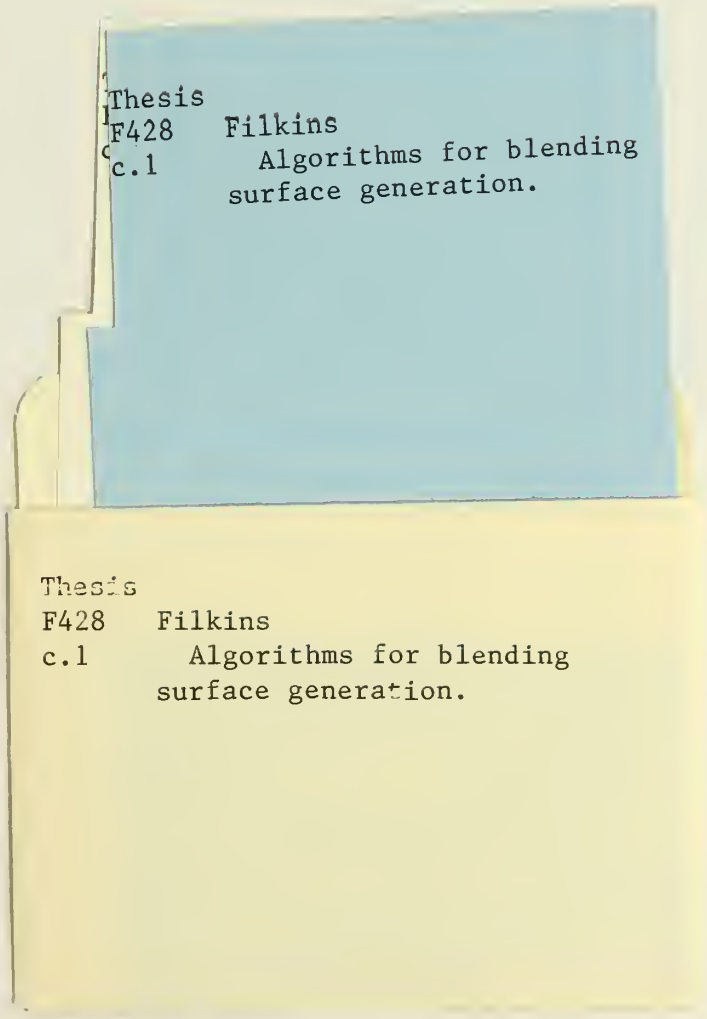
[Tuohy 91c] S. T. Tuohy and N. M. Patrikalakis. Geometric representation of marine propulsors. *Cambridge, Mass: MIT Ocean Engineering Design Laboratory Memorandum 91-7*, 1991.

[Varady 89] T. Varady, J. Vida, and R. R. Martin. Parameteric blending in a boundary representation solid modeller. *Mathematics of Surfaces 3, Clarendon Press, Oxford*, pages 171–197, 1989.

[Wolter 91] F.-E. Wolter and S. T. Tuohy. Approximation of high degree and procedural curves. *Engineering with Computers*, 1991. (To appear).

[Woodwark87] J. R. Woodwark. Blends in geometric modelling. *The Mathematics of Surfaces II, ed. R. R. Martin, Clarendon Press, Oxford*, pages 255–297, 1987.

497-769



Thesis
F428 Filkins
c.1 Algorithms for blending
surface generation.